



Deutsch

FUJITSU Software BS2000

CIS V12.0 Manual 5

Diagnosehilfe

Benutzerhandbuch

April 2020

## **Kritik... Anregungen... Korrekturen...**

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an [manuals@ts.fujitsu.com](mailto:manuals@ts.fujitsu.com) senden.

## **Zertifizierte Dokumentation nach DIN EN ISO 9001:2015**

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2015 erfüllt.

## **Copyright und Handelsmarken**

Copyright © 2020 Fujitsu Technology Solutions GmbH.

Alle Rechte vorbehalten. Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

## Inhaltsverzeichnis

1 Memory - Pool .....	9
1.1 Kommunikations - Pool.....	9
1.1.1 Allgemeines .....	9
1.1.2 Name des Common - Memory - Pools .....	9
1.1.3 Beschreibung des Poolanfangs (CISKOOR / CISDBH).....	10
1.1.3.1 Weitere Bereiche im Poolanfang (CISKOOR).....	13
1.1.3.2 Weiterer Pool (CISKOOR).....	14
1.1.3.3 Weitere Bereiche im Poolanfang (CISDBH).....	14
1.1.4 Beschreibung des Poolsatzes (CISKOOR / CISDBH) .....	15
1.1.4.1 Allgemeiner Kopf( DSECT POOLS0) .....	15
1.1.4.2 Datenteil (DSECT POOLS1).....	17
1.1.4.3 Beschreibung des PK - Satzes (CISDBH).....	22
1.1.4.4 Übersicht Kommandotypen .....	23
1.1.4.5 Benützung der Felder beim Senden.....	24
1.1.4.6 Benützung der Felder bei der Antwort.....	25
1.1.4.7 Bereich bei Kommandotyp 3 und 4.....	26
1.1.4.8 Bemerkung zur DCAM - Kommunikation.....	26
1.2 Pool der Dateinamen in CISKOOR .....	27
1.2.1 Allgemeines .....	27
1.3 Pool in CISUTM.....	28
1.3.1 Allgemeines .....	28
1.3.2 Aufbau.....	28
1.3.3 Kopfsatz .....	29
1.3.4 ZPV - Bereich .....	30
1.3.5 Verwaltungstabellen .....	30
1.3.6 Rio - Bereich .....	30
1.3.7 Arbeitsbereich in CISUTM .....	31
1.3.8 FCB - Infos.....	31
1.3.9 Parameter für CISUTM .....	32
1.4 Pool in CISKURZ.....	33
2 Kommunikation.....	35
2.1 Kommunikation CIS - CISKOOR.....	35
2.1.1 Übersicht.....	35
2.1.2 DV - technische Realisierung über P1 - Eventing .....	36
2.1.3 Namen der Datenwege (Event - Items):.....	37
2.1.4 Aufbau der Kommunikation .....	37
2.1.5 Kommunikation über P1 - Eventing .....	37
2.1.5.1 Prinzip .....	38
2.1.6 Kommunikation über Warteschlange .....	38
2.1.6.1 Prinzip .....	38
2.1.6.2 Elemente der Warteschlange .....	38
2.1.6.3 Errechnen der Adresse in der Warteschlange .....	39
2.1.6.4 Eintragen einer Information in die Warteschlange in CIS.....	39
2.1.6.5 Austragen einer Information aus der Warteschlange in CISKOOR .....	41
2.1.6.6 Beschreibung der Rückinformation .....	42
2.1.6.7 Senden der Rückmeldung in CISKOOR.....	43
2.1.6.8 Empfangen der Rückmeldung in CIS .....	43
2.1.6.9 Übersicht der Werte in POOLR1 .....	43
2.1.7 Abbau der Kommunikation .....	44
2.1.8 Zeitmessungen .....	44

2.2 Kommunikation CISCON - CISDBH .....	45
2.2.1 Allgemeines .....	45
2.2.2 Kommunikation über P1 - Eventing .....	45
2.2.3 Kommunikation über DCAM .....	45
2.2.3.1 Allgemeines .....	45
2.2.3.2 Aufbau der Kommunikation .....	45
2.2.3.3 Datenverkehr .....	46
2.2.3.3.1 Senden an CISDBH .....	46
2.2.3.3.2 Empfangen der Antwort.....	46
2.2.3.4 Abbau der Kommunikation .....	46
2.2.4 Zeitmessungen .....	47
2.2.4.1 Kommunikation über P1 - Eventing .....	47
2.2.4.2 Kommunikation über DCAM .....	47
2.2.5 Beschreibung des DBS - Feldes .....	48
3 Spezielle Dateien.....	51
3.1 Datensicherungsdateien.....	51
3.1.1 Allgemeines .....	51
3.1.2 Beschreibung der After - Image - Datei.....	53
3.1.2.1 Beschreibung der ersten PAM - Seite .....	53
3.1.2.2 Beschreibung eines After - Image - Satzes .....	54
3.1.2.3 Beispiel .....	56
3.1.2.4 Beschreibung eines Checkpoint - Satzes.....	57
3.1.3 Beschreibung der Before - Image - Datei.....	59
3.1.3.1 Beschreibung der ersten PAM - Seite .....	59
3.1.3.2 Beschreibung eines Before - Image - Satzes .....	61
3.1.4 Beschreibung der Protokolldatei.....	63
3.1.4.1 Beschreibung des ersten Satzes .....	63
3.1.4.1.1 Beschreibung eines Protokollsatzes .....	64
3.1.5 Beschreibung der Statusdatei .....	65
3.2 Datei in CISUTM.....	66
4 Tabellen.....	67
4.1 Tabellen in CISKOOR .....	67
4.1.1 KOOR - Task - Tabelle .....	67
4.1.2 Partnertabelle in CISKOOR.....	67
4.1.3 Beschreibung der TID - Tabelle in CISKOOR / CISBIM.....	70
4.1.4 Beschreibung der TID - Tabelle in CISSP .....	72
4.1.5 Beschreibung des Poolkopfes .....	73
4.1.6 Beschreibung des Pools der Sperrlisten in CISSP.....	74
4.2 Tabellen in CISDBH .....	77
4.2.1 DBH - Task - Tabelle .....	77
4.2.2 Partnertabelle in CISDBH .....	77
4.3 Tabellen in CISCON .....	80
4.3.1 Transaktionstabelle.....	80
4.3.2 DB - spezifischer Bereich im TAM.....	81
4.3.3 Paßwortabelle.....	81
4.3.4 DBH - Tabelle .....	82
4.3.4.1 Inhalt von DTDSAV bei Kommunikation über DCAM .....	83
4.3.4.2 Inhalt von DTDSAV bei Kommunikation über P1 - Eventing .....	84
4.3.5 Verkettung der Tabellen .....	85
4.3.6 Organisation der Tabellen im Speicher .....	86
4.3.7 Kommandotabelle in CISCON (Tabelle TKOM) .....	87
4.4 Tabellen in CISUTM .....	90
4.4.1 Tabelle der Transaktionen.....	90

5 Spezielle Kommandos.....	91	
5.1 Paßwortschutz.....	91	
5.2 Kommandos für CIS .....	92	
5.2.1 Übersicht.....	92	
5.2.2 CALLTRx	Protokollierung der CIS Aufrufe .....	93
5.2.3 CMEM	Speicherbelegung .....	94
5.2.4 CTIMEx	Zeitmessung.....	94
5.2.5 CTRx	SVC - Trace in CIS.....	95
5.2.6 CWRITE	Read - only Module überschreibbar machen.....	95
5.3 Kommandos für CISCON .....	96	
5.3.1 Übersicht.....	96	
5.3.2 CCDIA	Diagnose schalten.....	97
5.3.3 CCLST D	Auflistung der DBH - Tabelle .....	98
5.3.4 CCLST P	Auflistung der Paßworttabelle .....	99
5.3.5 CCLST T	Auflistung der Transaktionstabelle.....	100
5.3.6 CCMEM	Speicherbelegung .....	102
5.3.7 CCSYS	BS2000 Kommando .....	102
5.3.8 CCT=	Angabe einer TSN.....	102
5.4 Kommandos für CISUTM .....	103	
5.4.1 Übersicht.....	103	
5.4.2 UINFT	Info über Transaktionen .....	104
5.4.3 UDDIA	Dump der DIAGAREA.....	104
5.4.4 UDUMP	Dump des Programms .....	104
5.4.5 UERR	Fehlerabfangroutine .....	105
5.4.6 UTERMD	Dump und Programmende.....	105
5.4.7 USYS	BS2000 Kommando .....	105
5.4.8 UT=	Angabe einer TSN.....	105
5.5 Kommandos für CISKOOR.....	106	
5.5.1 Übersicht.....	106	
5.5.2 KT=	Angabe einer TSN.....	108
5.5.3 KCANc	Löschen eines Partnereintrags .....	108
5.5.4 KCANF	Löschen von Dateien in Pool 3 .....	109
5.5.5 KCANs	Löschen einer Statusinformation .....	109
5.5.6 KCANT	Rücksetzen einer Transaktion .....	110
5.5.7 KCRASH	Simulation eines Crashes .....	110
5.5.8 KDOFF	Diagnose ausschalten.....	111
5.5.9 KDON	Diagnose einschalten.....	111
5.5.10 KDUMP	Dump von CISKOOR .....	112
5.5.11 KECHO	Echo - Funktion .....	112
5.5.12 KENA	Steuerung von ENA - COMM.....	113
5.5.13 KEND	Setzen der Beendigungsbedingung.....	114
5.5.14 KINFB	Information über Before - Images .....	115
5.5.15 KINFC	Information über Partner .....	116
5.5.16 KINFF	Information über Dateien in Pool 3 .....	118
5.5.17 KINFL	Information über Sperren .....	120
5.5.18 KINFP	Information über Transaktionen in PTC .....	122
5.5.19 KINFS	Information über Stati.....	123
5.5.20 KINFT	Information über Transaktionen.....	124
5.5.21 KKILL	Dump und Ende des gesamten Systems .....	126
5.5.22 KMEM	Speicherbelegung .....	127
5.5.23 KNTASKS	Setzen der Anzahl Tasks .....	128
5.5.24 KPEND	PTC beenden .....	128
5.5.25 KPRESET	PTC rücksetzen.....	129
5.5.26 KSTAT	Ausgabe der Statistik ein/ ausschalten .....	129
5.5.27 KSYS	BS2000 Kommando ausführen.....	130
5.5.28 KTERMD	Dump und Ende von CISKOOR.....	131
5.5.29 KTIMEOFF	TIME ausschalten .....	131
5.5.30 KTIMEON	TIME einschalten .....	132
5.5.31 KTROFF	SVC - Trace ausschalten .....	132
5.5.32 KTRON	SVC - Trace einschalten .....	133
5.5.33 KWAIT	Warten.....	133
5.5.34 KWRITE	CISKOOR überschreibbar machen.....	134

5.6 Kommandos für CISDBH.....	135
5.6.1 Übersicht.....	135
5.6.2 DT=                   Angabe einer TSN.....	136
5.6.3 DCANC               Löschen eines Partnereintrags .....	136
5.6.4 DCL                 Datei(en) vorübergehend schließen .....	137
5.6.5 DCM                 CM der Tasks abfragen .....	137
5.6.6 DCMCL              CM der Tasks löschen .....	138
5.6.7 DDUMP              Dump aller DBH - Tasks .....	138
5.6.8 DENA               Steuerung von ENA - COMM.....	139
5.6.9 DEND               Setzen der Beendigungsbedingung von CISDBH .....	139
5.6.10 DINFC             Information über Partner .....	140
5.6.11 DINFT             Information über Transaktionen.....	143
5.6.12 DKILL             Dump und Ende des gesamten Systems .....	145
5.6.13 DNTASKS          Setzen der Anzahl Tasks .....	146
5.6.14 DOP               Datei(en) wieder eröffnen .....	147
5.6.15 DSEC              Datei(en) vorreservieren .....	147
5.6.16 DSTAT             Ausgabe der Statistik ein- und ausschalten.....	148
5.6.17 DTERMD           Dump und Ende von allen DBH - Tasks .....	148
5.6.18 DUNSEC           Vorreservierung auflösen.....	149
5.6.19 DWAIT            Warten.....	149
5.7 Kommandos für CISKURZ .....	150
5.7.1 Übersicht.....	150
5.7.2 Kommandokettung.....	151
5.7.3 %CISGEN            CISGEN Aufruf.....	151
5.7.4 %CTGET            CIS Passivkommando.....	152
5.7.5 %GET BUFFER       Unbenannte Variable aus EB setzen .....	153
5.7.6 %MODIFY BUFFER    EB setzen .....	153
5.7.7 %MODIFY LENGTH    LEB setzen .....	154
5.7.8 %PASSIV OFF       Passivbetrieb ausschalten .....	154
5.7.9 %PASSIV ON        Passivbetrieb einschalten .....	154
5.7.10 %SEND            Senden / Setzen bestimmter Ereignisse.....	155
5.7.11 %SHOW BUFFER    EB ausgeben.....	155
5.7.12 %SHOW LENGTH    EB ausgeben.....	156
5.7.13 SYS               BS2000 System Kommando .....	156
5.7.14 %WAIT            Warten auf bestimmte Ereignisse.....	157
5.7.15 %WHEN            Prüfen bestimmter Ereignisse.....	158
6 Erläuterungen zu den CO - Fehlermeldungen .....	159
6.1 Format der Fehlermeldungen.....	159
6.2 Übersicht über die Fehlermeldungen .....	159
6.3 Übersicht der Rückmeldungen von CISKOOR .....	160
6.4 Fehlermeldungen von CISAIM .....	161
6.5 Fehlermeldungen von CISBIM .....	162
6.6 Fehlermeldungen von CISPRO.....	163
6.7 Fehlermeldungen von CISSP .....	163
6.8 Hierarchie des Fehlermeldewesens .....	164
7 Diverse Beschreibungen zu CIS .....	165
7.1 Beschreibung der VMADL.....	165
7.2 Formatierter Auftrag .....	166
7.3 Beschreibung des Kommunikationsfeldes .....	168
7.4 Beschreibung des Zielpunktkopfes .....	169
7.5 Beschreibung des Zielpunktlistenkellers .....	170
7.6 Beschreibung eines Extraktionssatzes.....	171
7.7 Beschreibung eines VD - Satzes.....	171
7.8 Liste der Kommandos und Kennungen.....	172
7.9 Liste der Module und Kommandos.....	177

8 Diagnose - Routinen.....	183
8.1 Diagnose - Routinen in CISCON.....	183
8.1.1 Protokollierung der IUTMDB - Schnittstelle.....	183
8.1.1.1 Allgemeines.....	183
8.1.2 Einschalten der Protokollierung.....	183
8.1.3 Protokollierung von CISCON.....	184
8.1.4 UTM - DIAGAREA.....	187
8.2 Diagnose - Routinen in CISUTM.....	188
8.2.1 DIAGAREA.....	188
8.2.1.1 Kopf der DIAGAREA.....	188
8.2.1.2 Eintrag in der DIAGAREA.....	188
8.2.1.3 Auswertung der DIAGAREA.....	189
8.2.2 Fehlercodeabfrage in CISUTM.....	189
8.2.3 Protokollierung der CISUTM - Aufrufe.....	189
8.2.3.1 Allgemeines.....	189
8.2.4 Einschalten der Protokollierung.....	189
8.2.5 Protokollierung von CISUTM.....	190
8.2.5.1 Aufrufe.....	190
8.2.5.2 FCB - Infos.....	190
8.3 Diagnose - Routinen in CISUTMA.....	191
8.4 Diagnose - Routinen in CISDBH.....	192
8.5 Diagnose - Routinen in CISKOOR.....	192
9 Diverse DSECTS.....	193
10 Binderlisten.....	195





## 1.1.3 Beschreibung des Poolanfangs (CISKOOR / CISDBH)

Siehe auch Makro POOLA

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLADAT	0 - 5	6	C	Datum POOL - Initialisierung (JJNNN)
POOLAZEI	6 - B	6	C	Uhrzeit POOL - Initialisierung (HHMMSS)
POOLANR	C - F	4	F	Laufende Nummer (In CISKOOR benützt zum Numerieren der AIM - , BIM - , und PRO - Sätze).
POOLAZK	10 - 13	4	C	Zustand CISKOOR/CISDBH:  OK      CISKOOR/CISDBH o.k.  OKLE    CISKOOR/CISDBH o.k aber ENA - COMM gesperrt.  ENDE    CISKOOR/CISDBH beendet.  LADE    CISKOOR/CISDBH wird geladen.  RST      CISKOOR Restart läuft.  FRST    CISKOOR Fehler bei Restart.  KILL     CISKOOR/CISDBH "gekillt".
POOLAZA	14 - 15	2	H	Anzahl aktueller Tasks.
POOLAZM	16 - 17	2	H	Maximale Anzahl Tasks.
POOLAZV	18 - 19	2	H	Vorgegebene Anzahl Tasks.
	1A - 1B	2		frei
POOLATA	1C - 1F	4	A	Anfangsadresse TSN - Tabelle im Pool.
POOLATP	20 - 23	4	A	Pointer zum 1. freien Eintrag.
POOLATE	24 - 27	4	A	Endadresse TSN - Tabelle im Pool.
POOLAKE	28 - 28	1	C	Beendigungsbedingung: 0, 1, 2, 3
POOLAKES	29 - 29	1	C	Gesicherte Beendigungsbedingung für Zyklen über 24 Stunden hinaus.
POOLAP	2A - 2A	1	C	Kennung für "Primary" V        : "Verwaltungstask" Space    : "Normaler" Task
POOLASVC	2B - 2B	1		Flag für SVC - Trace.
POOLATIM	2C - 2C	1		Flag für Time - Liste.
	2D - 2D	1		frei

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLACS	2E - 37	A		Für CIS - Spezial - Kommando, das an die anderen CISDBH - Tasks dieser DBH - ID geschickt werden soll.
POOLACSK	2E - 31	4	X	Kennungen aus formatiertem Kommando: - Modul - Nr.   Byte 8 in Kopf - Operation     Byte 9 in Kopf - UP - Nr.       Byte A in Kopf - Typ            Byte B in Kopf
POOLACSP	32 - 37	6	C	CIS - Paßwort
POOLATS	38 - 3B	4	N	TSN, die das Kommando sendet.
	3C - 3F	4		frei
POOLALME	40 - 43	4	F	MEM - Wert (CISSIL - nur CISKOOR)
POOLALTI	44 - 47	4	F	TIM - Wert (CISSIL - nur CISKOOR)
POOLALTS	48 - 4B	4	F	TSK - Wert (CISSIL - nur CISKOOR)
	4C - 6B	32		frei
POOLAWZ	6C - 6F	4	F	Warteschlangenzähler: - CIS/CISCON       : + 1 - CISKOOR/CISDBH : - 1
POOLAWZI	70 - 73	4	F	Initialwert/Vergleichswert für POOLAWZ (Anzahl Tasks).
POOLAKV	74 - 77	4	C	Kennungen über Kommunikation.
POOLAKV1	74 - 75	2	C	Richtung "hin" CIS → CISKOOR/CISCON → CISDBH - WS über Warteschlange - Spaces über P1 - Eventing
POOLAKV2	76 - 77	2	C	Richtung "zurück" CISKOOR → CIS/CISDBH → CISCON - WS über Warteschlange - Spaces über P1 - Eventing
POOLAWAE	78 - 7F	4	F	Anzahl Einträge in Warteschlange (500 = X'1F4').
POOLAWPC	80 - 83	4	F	Position von CIS/CISCON in der Warteschlange (fortlaufend nummeriert).
POOLAWPK	84 - 87	4	F	Position von CISKOOR/CISDBH in der Warteschlange (fortlaufend nummeriert).
POOLAWS	88 - 857	7D0 2000	F	Warteschlange. 500 Einträge. Jeder Eintrag ist 1 Wort mit Inhalt: F'0'       frei F'd'       d = Distanz zu POOL - Anfang des POOL - Satzes. X'FF0iiii'   iiii = Id. der Kommunikation Format für Disable Comm.

Memory - Pool

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLAWSE	858 - 858			(Ende + 1)
POOLAIA	858 - 85B	4	F	Anfangsadresse TID - Tabelle (Distanz zu Poolanfang)
POOLAIP	85C - 85F	4	F	Pointer 1. freier Eintrag TID - Tabelle (Distanz zu Poolanfang)
POOLAIE	860 - 863	4	F	Endadresse TID - Tabelle (Distanz zu Poolanfang)
POOLASA	864 - 867	4	F	Anfangsadresse STA - Tabelle (Distanz zu Poolanfang)
POOLASP	868 - 86B	4	F	Pointer 1. freier Eintrag STA - Tabelle (Distanz zu Poolanfang)
POOLASE	86C - 86F	4	F	Endadresse STA - Tabelle (Distanz zu Poolanfang)
	870 - FFF	790 1936		frei

### 1.1.3.1 Weitere Bereiche im Poolanfang (CISKOOR)

Ab der 2. Seite des Pools stehen Tabellen, die allen Tasks dieses CISKOORs (Multi - Task - Betrieb) zur Verfügung stehen müssen. Dies sind:

#### 1. Die CISKOOR - Tabelle

Für jeden möglichen Task für diesen Multi - Task CISKOOR gibt es einen Eintrag von 8 Bytes. Die ersten 4 Bytes enthalten die TSN des Tasks (oder Spaces wenn es den Task noch nicht gibt), die nächsten 4 Bytes werden als Rückmeldung für die Aufrufe von CISKOOR - Task zu CISKOOR - Task benützt.

Die Anzahl der Einträge ist gleich dem Wert aus dem CISKOOR - Parameter KMAXT=.

#### 2. Die Partnertabelle

Für jeden möglichen Partner gibt es einen Eintrag in dieser Tabelle. Die Anzahl der Einträge ist gleich dem Wert aus dem CISKOOR - Parameter KNCOMM=.

Jeder Eintrag hat einen festen Teil von 48 Bytes und einen variablen Teil mit Feldern von 8 Bytes. Die Felder enthalten die Kurzbezeichnung und Referenznummer der Kommunikation zu dem Partner für einen CISKOOR - Task; deshalb gibt es so viele 8 - Byte Felder wie es maximal CISKOOR - Tasks geben kann (Wert aus KMAXT=).

#### 3. Die Transaktionstabelle

Für jede mögliche Transaktion gibt es einen Eintrag in dieser Tabelle. Die Anzahl der Einträge ist gleich dem Wert aus dem CISKOOR - Parameter KNTRANS=.

Anfangsadresse, Pointer und Endadresse stehen als Distanz zum Poolanfang in den Feldern POOLAIA, POOLAIP und POOLAIE.

#### 4. Die Statustabelle

Für jeden möglichen Statuseintrag gibt es einen Eintrag in dieser Tabelle. Die Anzahl der Einträge ist gleich dem Wert aus dem CISKOOR - Parameter KNSTA=.

Anfangsadresse, Pointer und Endadresse stehen als Distanz zum Poolanfang in den Feldern POOLASA, POOLASP und POOLASE.

Die Beschreibungen dieser Tabellen befinden sich im Kapitel "Tabellen". (Vgl. Seite 67 ff)

### 1.1.3.2 Weiterer Pool (CISKOOR)

CISKOOR legt noch Bereiche für CISAIM, CISBIM, CISSP an. Läuft CISKOOR als Monotask, so werden die Bereiche im Klasse - 6 - Speicher angelegt. Im Multitask - Betrieb wird ein eigener Memory - Pool angelegt. Der Name unterscheidet sich nur durch eine 2 an Byte + 22.

Inhalt dieses Bereichs:

#### 1. Sperr - Bereiche

Bestehen aus einem Kopf (64 Bytes), den Verwaltungseinträgen (KNLOCK Einträge @ 116 Bytes) und dem eigentlichen Sperr - Pool in der Größe von KLOCKP.

#### 2. AIM - Bereiche

Bestehen aus 1. PAM - Page, 2. PAM - Page und Puffer für AIM - Satz von 36K.

#### 3. BIM - Bereiche

Bestehen aus 1. PAM - Page, gemeinsamen Bereichen von 2K, 2K - Puffer für Statussätze und Puffer für BIM - Satz von 36K.

### 1.1.3.3 Weitere Bereiche im Poolanfang (CISDBH)

Ab der 2. Seite des Pools stehen zwei Tabellen, die allen Tasks dieses DBH's (Multi - Task - Betrieb) zur Verfügung stehen müssen.

#### 1. Die DBH - Tabelle

Für jeden möglichen Task des Multi - Task - DBH gibt es einen Eintrag von 8 Bytes. Die ersten 4 Bytes enthalten die TSN des Tasks (oder Spaces wenn es den Task noch nicht gibt), die nächsten 4 Bytes werden als Rückmeldung für die Aufrufe von DBH - Task zu DBH - Task benützt.

Die Anzahl der Einträge ist gleich dem Wert aus dem CISDBH - Parameter DMAXT=.

#### 2. Die Partnertabelle

Für jeden möglichen Partner gibt es einen Eintrag in dieser Tabelle. Die Anzahl der Einträge ist gleich dem Wert aus dem CISDBH - Parameter DNCOMM=.

Jeder Eintrag hat einen festen Teil von 48 Bytes und einen variablen Teil mit Feldern von 8 Bytes. Die Felder enthalten die Kurzbezeichnung und Referenznummer der Kommunikation zu dem Partner für einen DBH - Task; deshalb gibt es so viele 8 - Byte Felder wie es maximal DBH - Tasks geben kann (Wert aus DMAXT=).

Die Beschreibung dieser Tabelle befindet sich im Kapitel "Tabellen". (Vgl. Seite 77)

## 1.1.4 Beschreibung des Poolsatzes (CISKOOR / CISDBH)

### 1.1.4.1 Allgemeiner Kopf( DSECT POOLS0)

Siehe auch Makro POOLS

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLS0L	0 - 1	2	H	Länge dieses Kopfes.
POOLS0V	2 - 3	2	H	Version des Senders (11).
POOLID1	4 - 5	2	H	ID für diese TSN.
POOLID2	6 - 7	2	H	Version des Senders (11).
POOLK1	8 - B	4	F	1 - Enable communication. 2 - Disable communication. 3 - Schreiben AIM. 4 - Schreiben BIM. 5 - Schreiben PRO. 6 - Schreiben checkpoint (TR). 7 - Sperren 8 - Entsperren 9 - Diagnose (\$D), auch Kommando für CISDBH. A - Sperrabfragen
POOLK2	C - F	4	F	1 - Elim 2 - Store 3 - Insert
POOLR1	10 - 13	4	F	Return - Code - 1 00 - Noch nicht verarbeitet FE - Disret verarbeitet FF - Verarbeitet, alles o.k. C0 - CIS/CISCON wartet auf Quittung. 01 - POOLKL1 falsch. 02 - POOLK2 falsch. 03 - Fehler bei Schreiben AIM. 04 - Fehler bei Schreiben BIM. 05 - Fehler bei Schreiben PRO. 06 - Fehler bei Schreiben Checkpoint. 07 - Fehler bei Sperren. 08 - Fehler bei Entsperren. 09 - Fehler bei Diagnose. 0A - Fehler bei Abfrage Sperre.
POOLR2	14 - 17	4	F	Return - Code - 2 00 - Noch nicht verarbeitet. FF - O.k.  oder Fehlernummer.

Memory - Pool

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLS0N1	18 - 1B	4	F	Nummer - 1 (Reserve)
	1C - 1F	4	F	Reserve
POOLS001	20 - 27	8	D	Zeit - 1 (STCK - Format)
POOLS003	28 - 2F	8	D	Zeit - 3 (STCK - Format)
POOLS012	30 - 33	4	F	Zeit - 1 - 2 (Mikrosekunden)
POOLS023	34 - 37	4	F	Zeit - 2 - 3 (Mikrosekunden)
POOLS034	38 - 3B	4	F	Zeit - 3 - 4 (Mikrosekunden)
POOLS014	3C - 3F	4	F	Zeit - 1 - 4 (Mikrosekunden)

### 1.1.4.2 Datenteil (DSECT POOLS1)

Siehe auch Makro POOLS

Ab hier werden die Daten in die AIM - , BIM- und PRO - Datei geschrieben.

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLLB	0 - 1	2	H	Länge des Satzes ab hier.
	2 - 3	2		Spaces
POOLKL	4 - 5	2	H	Länge des "Kopfes".
	6 - 7	2		Spaces
POOLDAT	8 - D	6	C	Datum der POOL - Initialisierung im Format JJNNN
POOLZEIT	E - 13	6	C	Uhrzeit der POOL - Initialisierung im Format HHMMSS
POOLNR	14 - 17	4	F	Fortlaufende Nummer (aus POOLANR).
POOLTYP	18 - 18	1	C	Typ (siehe Makro POOLS).
POOLTYP2	19 - 19	1	C	Zusatztyp (siehe Makro POOLS).
POOLCH1	1A - 1A	1	C	Checkpointstatus (siehe Makro POOLS).
POOLCH2	1B - 1B	1	C	Checkpointkennung (siehe Makro POOLS).
POOLCH3	1C - 1C	1	C	Transaktion o.k ? (siehe Makro POOLS).
POOLBETR	1D - 1D	1	C	Betriebsart: D = CISDBH N = TIAM S = UTM - synchronisiert U = UTM - unsynchronisiert
POOLTSN	1E - 21	4	C	TSN
POOLTID	22 - 29	8	C	Transaktions - id.
POOLAPPL	2A - 31	8	C	Applikationsname
POOLHOST	32 - 39	8	C	Hostname
POOLDIV	3A			Ab hier verschieden je nach Typ.
Bei enable communication (CISKOOR):				
POOLMPS	3A - 3D	4	F	P1 - Eventing: Poolgröße in Multiplen von 4K
POOLEUI	3E - 45	8	C	User - id.
POOLEUS	46 - 4D	8	C	UTM - user

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
Bei enable communication (CISDBH):				
POOLMPS	3A - 3D	4	F	P1 - Eventing: Poolgröße in Multiplen von 4K. DCAM: 2 Bytes max. Länge CISCON (4096). 2 Bytes max. Länge DCAM.
POOLEUI	3E - 45	8	C	User - id.
POOLEUS	46 - 4D	8	C	UTM - user
	4E - CB	7E		Nicht benützt.
	CC -			Liste der Paßworte (Rückmeldung).
	CC - CD	2	H	- Länge (8*p*7)
	CE - CE	1	C	- Zustand CISKOOR.
	CF - CF	1		- Space
	D0 - D3	4	F	- Anzahl Paßwörter (F'p'). Wenn keine Paßworte, dann p = 0.
	D4 -	p*7	C	- p Paßworte à 6 + 1 Bytes: 6 Bytes Paßwort 1 Byte Typ (D oder J) mit: $0 \leq p \leq 512$
Bei Protokollsatz schreiben / Checkpoint schreiben / Diagnose:				
POOLPUI	3A - 41	8	C	User - id. (CISKOOR)
POOLKCUS	3A - 451	8	C	UTM - user (CISDBH)
POOLPZT	42 - 47	6	N	Uhrzeit des Senders (HHMMSS)
POOLDPL	48 - 49	2	H	Länge der folgenden Daten. (Bei Checkpoint schreiben bleiben die folgenden Felder unbenützt.)
	4A - 4B	2		Spaces
POOLDP	4C -			Daten: - Protokollsatz oder - Kommando bei \$D.
Bei Schreiben AIM / Schreiben BIM / Sperren:				
POOLDN	3A - 6F	54	C	Vollqualifizierter Dateiname.
POOLRF	70 - 70	1	C	RECFORM: x'04' = F x'02' = V
POOLKLEN	71 - 71	1	B	KEYLEN - 1
POOLKLOC	72 - 73	0	H	KEYLOC - 1
POOLSPK	74 - 74	1	C	Kennung bei Sperren:  C'A' Ordnungsbegriffe aufsteigend. (Original - ZPL) C'R' Ordnungsbegriffe Random. (Sortierte ZPL) C'V' Ordnungsbegriffe aufsteigend. (Verdichtete ZPL)
POOLBLKS	74 - 74	1	X	Blocksize bei AIM

Feld	Bytes - hex.	Länge hex.	Typ	Inhalt
POOLBK	75 - 75	1	C	BIM - Kennung: C'1' Erstes Mal (CISKOOR prüft ob Datei erreichbar) C'2' Weitere Male
POOLOPEN	75 - 75	1	X	OPEN - Typ bei AIM: X'40' OUTIN X'20' INOUT X'04' OUTPUT
POOLDL	76 - 77	2	H	Länge Datenteil.
	78 - 79	2		Spaces
POOLD	7A -			Datenteil: - AIM - Satz - BIM - Satz - X - Satz
Bei CIS - Kommando (CISCON $\longleftrightarrow$ CISDBH)				
POOLKCUS	3A - 41	8	C	UTM - user (CISDBH)
POOLPZT	42 - 47	6	N	Uhrzeit des Sendens (HHMMSS)
POOLDPL PKSL	48 - 49	2	H	Länge der folgenden Daten.
	4A - 4B	2		Spaces
PKTYP . . .	4C - ...			Siehe PK - Satz

Beschreibung der Kennungen im POOL von CISKOOR

Es folgt eine Liste der möglichen Kennungen im POOL von CISKOOR. Die POOL - Felder tragen in der Liste die Titel:

POOLK1: K1  
 POOLK2: K2  
 POOLTYP: Typ - 1  
 POOLTYP2: Typ - 2  
 POOLCH1: Typ - 3  
 POOLCH2: Typ - 4  
 POOLCH3: Typ - 4

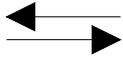
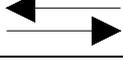
K1	K2	Funktion	Typen					Bemerkung/Erläuterung	CIS - Kommando
			1	2	3	4	5		
1		ENA							
2		DIS							
3	1	AIM	A	E				Elim	
	2		A	S				Store	
	3		A	I				Insert	
	4		A	X				X - Satz	
4	1	BIM	B	E				Elim	
	2		B	S				Store	
	3		B	I				Insert	
	4		B	X				X - Satz	
	5		B	P				Prüfen + Store	
5	1	PRO	P	P				Protokollierung	
	2		P	D				Diagnose	
6	1	CHK	?	C	E	A		C - expl. Anfang	TR,A
	2		?	C	E	E		C - expl. Ende	TR,E
	3		?	C	I	A		C - impl. Anfang	
	4		?	C	I	E		C - impl. Ende	
	5		?	C	R			C - reset TSN	
	6		?	C	B			C - reset TID	TR,R
	7		?	\$	A			Statusabfrage	\$@#TR,UA
	8		?	\$	L			Stati löschen	\$@#TR,UL
	9		?	C	E	L		C - expl. Anfang - Lesen	TR,AL
	10		?	C	E	P		UTM - Peta	\$@#TR,UP
	11		?	C	U	E		UTM - Warmstart Ende	\$@#TR,UE
	12		?	C	U	R		UTM - Warmstart Reset	\$@#TR,UR
	13		?	\$	C			Status CISKOOR	\$@#TR,UC
	14		?	C	W	P		Extern - Peta	TR,PE
	15		?	C	W	A		Extern - Statusabfrage	TR,SA
	16		?	C	W	L		Extern - Status löschen	TR,SL
	17		?	C	W	E		Extern - Warmstart Ende	TR,WE
	18		?	C	W	R		Extern - Warmstart Reset	TR,WR
	19		?	C	E	B		C - expl. Anfang + Stati führen.	TR,AS

K1	K2	Funktion	Typen					Bemerkung/Erläuterung	CIS - Kommando
			1	2	3	4	5		
7	1	SPA	S	L		A		Liste	
	2		S	S		A		Satz	
	3		S	D		A		Datei	
	4		S	R		A		Reserve	
	5		S	V		A		Secure (Vorreservieren)	
8	1	SPE	S	L		E		Liste	
	2		S	I		E		TID	
	3		S	T		E		TSN	
	4		S	R		E		Reserve	
	5		S	V		E		Secure	
9	1	DIA	D					\$D	
10	1	SPF	S	F	S	D		Sätze/Datei	ST,SD
	2		S	F	S	T		Sätze/TID	ST,ST
	3		S	F	S	A		Sätze/Datei + TID	ST,SA
	4		S	F	S	G		Sätze gesamt	ST,SG
	5		S	F	D	T		Dateien/TID	ST,DT
	6		S	F	D	G		Dateien gesamt	ST,DG
	7		S	F	D	R		Dateien reserviert	ST,DR

? steht für A: Nur After - Images  
 B: Nur Before - Images  
 C: Beides

## 1.1.4.3 Beschreibung des PK - Satzes (CISDBH)

Siehe auch Makro PKS

Feld	Bytes im POOL	Länge hex. dez.	Typ	Inhalt	Richtung
PKSL	48 - 49	2	H	Satzlänge	
	4A - 4B	2		Spaces	
PKTYP	4C - 4C	1	C	Kommando - Typ (Siehe nächste Seite)	
PKDBS	4D - 4D	1	C	DB - Statusfeld	
PKAP	4E - 4E	1	C	Aktiv- / Passiv - Kennung	
	4F - 50	2		reserviert	
PKACK	51 - 51	1	X	Kennung ob Accounting - Info	
PKAC	52 - 69	18 24	F	Accounting - Infos (UTM - Format: 6 Worte)	
PKKLA	6A - 6B	2	H	Länge Kommando auf Wortgrenze aufgerundet	
PKKL	6C - 6F	4	F	KL	
PKAZI	70 - 73	4	F	AZI / LZI	
PKCM	74 - 77	4	C	CM	
PKVM	78 - C7	50 80	C	VM	
PKZI	C8 - CB		F	ZI	
PKK	CC -		F	K	
				A / KAP / Bereich	

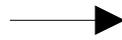
 Feld wird in Richtung CISCON nach CISDBH benützt.

 Feld wird in Richtung CISDBH nach CISCON benützt.

### 1.1.4.4 Übersicht Kommandotypen

Diese Typen werden in CISCON benützt.

Typ	Anzahl Parameter	5. Parameter		6. Parameter	
		Inhalt	Richtung	Inhalt	Richtung
0	4				
1	5	AZI/ASK	←		
2	5	A/KAP	→		
3	5	Bereich	←		
4	5	Bereich	→		
		Bereich	←		
5	6	LZI	→		
		LZI	←	ZI	←

 von CISCON nach CISDBH  
 von CISDBH an CISCON zurück.

## 1.1.4.5 Benützung der Felder beim Senden

Richtung CISCON nach CISDBH

Feld	Länge	Inhalt des Feldes je nach Typ					
		Typ = 0	Typ = 1	Typ = 2	Typ = 3	Typ = 4	Typ = 5
PKSL	2	132+k $\ell$ a	132+k $\ell$ a	132+k $\ell$ a+l	132+k $\ell$ a	132+k $\ell$ a+l	132+k $\ell$ a
PKTYP	1	0	1	2	3	4	5
PKDBS	1	Bisheriger Wert des DBS - Feldes.					
PKAP	1	"A" für Aktiv, "P" für Passiv.					
PKACK	1	"J" wenn accounting eingeschaltet.					
PKAC	24						
PKKLA	2	Länge des Kommandos aufgerundet. (k $\ell$ a)					
PKKL	4	Länge des Kommandos. (k $\ell$ )					
PKAZI	4		Inhalt von AZI / ASK				Inhalt von LZI
PKCM	4						
PKVM	80						
PKZI	4						
PKK	k $\ell$ a	Kommando					
	$\ell$			A / KAP		Bereich	

Erläuterung: k $\ell$  = Länge des CIS - Kommandos.  
k $\ell$ a = Länge des CIS - Kommandos aufgerundet auf ein Vielfaches von 4.  
 $\ell$  = Länge von A/KAP/Bereich (immer V - Format, Länge im 1. Halbwort).

Nicht beschriebene Felder werden nicht benützt.

## 1.1.4.6 Benützung der Felder bei der Antwort

Richtung CISDBH zurück an CISCON

Feld	Länge	Inhalt des Feldes je nach Typ					
		Typ = 0	Typ = 1	Typ = 2	Typ = 3	Typ = 4	Typ = 5
PKSL	2	128	128	128	128+ $\ell$	132+k $\ell$ a+ $\ell$	128+ $\ell$
PKTYP	1	0	1	2	3	4	5
PKDBS	1	Neuer Wert des DBS - Feldes					
PKAP	1	"A" / "P"					
PKACK	1	"J" / "N"					
PKAC	24	Accounting - Infos für UTM wenn PKACK = "J"					
PKKLA	2					k $\ell$ a	
PKKL	4						
PKAZI	4	AZI / ASK					Länge der Daten ( $\ell$ )
PKCM	4	Codierte Meldung					
PKVM	80	Verbale Meldung (nur wenn CM ungleich IM ...)					
PKZI	4 $\ell$				Bereich ( $\ell$ )	4 Bytes Reserve	ZI (Länge $\ell$ )
PKK	k $\ell$ a					k $\ell$ a Bytes Reserve	
	$\ell$					Bereich ( $\ell$ )	

Erläuterung: k $\ell$ a = Länge des CIS - Kommandos aufgerundet. $\ell$  = Länge von A / KAP / Bereich (immer V - Format, Länge im 1.Halbbyte).

Nicht beschriebene Felder werden nicht benützt.

**1.1.4.7 Bereich bei Kommandotyp 3 und 4**

Der Bereich enthält eine auszugebende Nachricht. Format:

Bytes	Länge	Inhalt
0 - 1	2	Länge insgesamt ( = Länge Text + 8 )
2 - 3	2	Spaces
4 - 7	4	<p>a) x'FF000000' wenn ein formatierter Bildschirm schon aufgebaut ist.</p> <p>b) x'000000ll' wenn Zeilen mit der Länge ll auszugeben sind. Die Anzahl der Zeilen errechnet sich aus:</p> $\text{Datenlänge} / ll$ <p>Bei jeder Ausgabe kann ll einen anderen Wert annehmen.</p> <p>c) x'00zz00ll' wenn Zeilen auszugeben sind. Jede Zeile hat die Länge ll. Es soll jedoch nur bis zum Zeichen zz ausgegeben werden und zwar jede Zeile mit einem eigenen Ausgabebefehl.</p> <p>d) x'F000llll' wenn nur eine Zeile der Länge llll auszugeben ist.</p> <p>e) x'F0zzllll' wenn nur eine Zeile der Länge llll auszugeben ist. Es soll jedoch nur bis zum Zeichen zz ausgegeben werden.</p>
8 - (7*X)	X	<p>Text:</p> <p>a) Formatierter Text in der Länge wie in Bytes 0 - 1 minus 8.</p> <p>b) 0 - n Zeilen der Länge wie in Byte 7 angegeben.</p> <p>c) 0 - n Zeilen der Länge wie in Byte 7 angegeben. In jeder Zeile steht an einer Stelle zwischen 1 und ll das Zeichen zz.</p> <p>d) 1 Zeile der Länge wie in Bytes 6 - 7 angegeben.</p> <p>e) 1 Zeile der Länge wie in Bytes 6 - 7 angegeben. An einer Stelle zwischen 1 und llll steht das Zeichen zz.</p>

**1.1.4.8 Bemerkung zur DCAM - Kommunikation**

Die Kommunikation zwischen CISCON und CISDBH kann über DCAM ausgeführt werden.

Der Satz, der von CISCON an CISDBH übergeben und von CISDBH an CISCON zurückgesendet wird, ist der Poolsatz bestehend aus den Teilen POOLS0 (X'40' = 64 Bytes) und POOLS1 bis Feld POOLDPL (X'48' = 72 Bytes) und PK - Satz.

Die Länge der Daten kann aus dem Inhalt von PKSL plus X'88' (=136) (in beiden Richtungen) errechnet werden.

## 1.2 Pool der Dateinamen in CISKOOR

### 1.2.1 Allgemeines

Alle CISKOOR - Tasks haben Zugriff auf einen Memory - Pool, in dem die von den verschiedenen CISKOORs bearbeiteten Dateien stehen. So kann sichergestellt werden, daß eine bestimmte Datei nicht in mehr als einem CISKOOR bearbeitet wird.

Eine Datei wird mit dem CISKOOR - Kennzeichen (KOORID) eingetragen wenn das erste Mal für sie in einem CIS - Task ein Before - Image Satz geschrieben wird. Hier wird auch geprüft, ob die Datei nicht schon von einem anderen CISKOOR eingetragen wurde.

Die Dateien werden ausgetragen wenn sich der letzte Task einer CISKOOR - Kennung beendet.

Dieses System erscheint vielleicht etwas starr, man bedenke aber, daß es nur eine Kontrolle ist. Der Betrieb ist so zu organisieren, daß es für jede Datei einen einzigen zuständigen CISKOOR gibt.

Name des Memory - Pools:

@XXXX@CISKOOR@CMEMPOOL3@C12

Der Memory - Pool hat eine Größe von 1 MB. Er hat folgenden Inhalt:

Bytes 0 - 3: Distanz zu freiem Eintrag  
Bytes 4 - 7: Anzahl Bytes im Pool  
ab Byte 8: Einträge: 1 Byte: KOORID (Space = ohne KOORID)  
1 Byte: Länge Dateiname - 1  
x Bytes: Dateiname

## 1.3 Pool in CISUTM

### 1.3.1 Allgemeines

CISUTM benützt einen Memory - Pool um Daten zwischen den im Multitaskbetrieb laufenden Tasks auszutauschen. Dieser Memory - Pool muß in jedem Task an der gleichen Adresse stehen. Deshalb wird er mit dem Parameter FIXED=YES angelegt. Ist in CISVARI+2A8 ein Wert eingetragen, so wird der Memory - Pool an die hier angegebene Adresse gelegt. Diese Adresse muß in jedem Task der Anwendung gleich sein. Außerdem soll diese Adresse auf Megabyte - Grenze liegen.

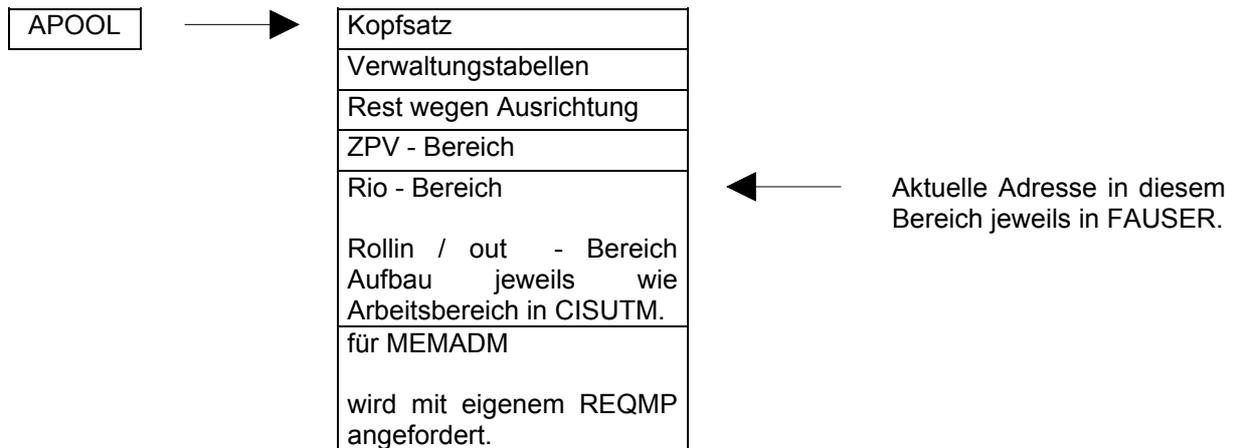
Der Name des Memory - Pools ist 24 Bytes lang hat folgendes Format:

CISUTM@CMEMPOOL@xxxxxxxx

wobei xxxxxxxx der Name der Anwendung ist. Der SCOPE - Wert ist GROUP.

### 1.3.2 Aufbau

Der Memory - Pool in CISUTM hat folgenden Aufbau:



APOOL = CISUTM + X'37D4'

## 1.3.3 Kopfsatz

Bytes	Länge	Inhalt
0 - 3	4	Distanz zu Verwaltungstabellenanfang
4 - 7	4	Distanz zu Verwaltungstabellenende
8 - B	4	Distanz zum ersten freien Eintrag in den Verwaltungstabellen
C - F	4	Anzahl Einträge in Verwaltungstabelle
10 - 13	4	Distanz zum ZPV - Bereich
14 - 17	4	Distanz zum Rio - Bereich - Anfang
18 - 1B	4	Distanz zum Rio - Bereich - Ende
1C - 1F	4	Distanz zum MEM - ANKER
20 - 27	8	Kennungen: 20 - Rio - Datei löschen am Ende? 21 - ZPL - Datei löschen am Ende? 22 - Trace - Flag 23 - Time - Flag 24 - SVCTRC - Flag Rest nicht benutzt.
28 - 2B	4	Zustand: O. K. DUMP TRD (=TERMD)
2C - 2F	4	Dump - Nummer (gepackt)
30 - 83		MEM - ANKER 5 Felder mit jeweils: 12 Bytes x'FF' 4 Bytes gepackte 0 1 Feld mit: x'FEFEFEFE' als Endekennung.
84	4	Adr. - Adr. 1. Datenbeschreibung

Memory - Pool

### 1.3.4 ZPV - Bereich

Der ZPV - Bereich liegt vor den Transaktionsbereichen im Memory - Pool von CISUTM. Es ist der Speicherbereich, in dem die Verwaltung der Zielpunktlisten liegt.

Die Zielpunktlisten selbst stehen in der ZPL - Datei mit dem Namen:

CIS.ZPL.aaaaaaaa (aaaaaaaa = Anwendungsname)

Diese Datei wird als PAM - Datei, sofern sie noch nicht vorhanden ist, per FILE- Kommando mit:

SPACE=(96,96) und BLKSIZE=(STD,16)

eingrichtet. Sie enthält Blöcke à 2K, die ab 1 durchnummeriert werden.

Der ZPV - Bereich hat Einträge von 4 Bytes. Das letzte Wort des Bereichs dient zur Feststellung, um welchen Task es sich handelt:

F'0' bedeutet 1.Task.

F'1' bedeutet, daß es weitere Tasks sind, d.h. der Memory - Pool und die PAM - Datei existieren schon.

Beschreibung eines Eintrags:

Bytes	Länge	Inhalt
0 - 1	2	Logische Dateinummer (als 2er Komplement) oder x'0000' wenn nicht benützt. FFFF = Datei Nr. 1 FFFE = Datei Nr. 2 ...
2 - 3	2	Logische Blocknummer (als 2er Komplement) oder x'0000' wenn nicht benützt.

Der 1. Eintrag entspricht der 1.PAM - Page in der Datei usw.. Die Größe des ZPV - Bereichs wird von dem Feld KAZPV bestimmt.

### 1.3.5 Verwaltungstabellen

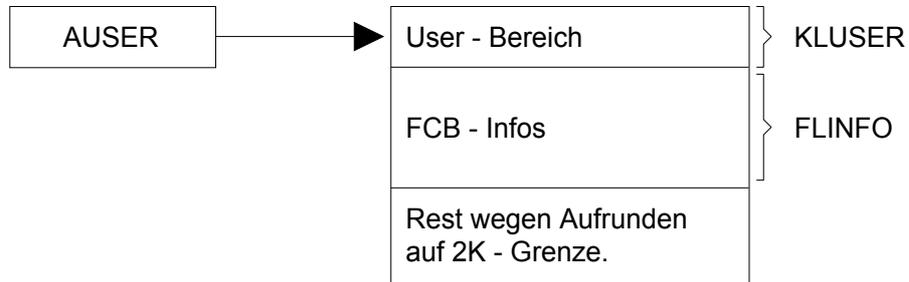
Diese sind im Kapitel "Tabellen" (vgl. Seite 67 ff) beschrieben.

### 1.3.6 Rio - Bereich

Für jeden im Speicher zu führenden Transaktionsbereich ist hier Platz vorhanden. Der Aufbau ist wie im Kapitel "Arbeitsbereich" (vgl. Seite 31) beschrieben. Die Anzahl Einträge kommt aus KRIOANZ bzw. DNRIO=

### 1.3.7 Arbeitsbereich in CISUTM

Es wird folgender Bereich angelegt:



AUSER = CISUTM + X'37D0'  
 KLUSER = CISUTM + X'3558'  
 FLINFO = CISUTM + X'37B0'  
 FAUSER = CISUTM + X'39FC'

Dieser Bereich wird benutzt, wenn die Daten in der PAM - Datei ausgelagert sind. Dann wird AUSER nach FAUSER gespeichert. Die Daten werden in die Datei geschrieben bzw. aus der Datei gelesen.

### 1.3.8 FCB - Infos

Diese stehen immer gleich hinter dem User - Bereich. (FAUSER = Adresse User - Bereich. FLUSER = Länge dieses Bereichs). Der Bereich ist so groß, daß Platz für alle zugelassenen FCB's ist (plus 12 Bytes). Die Anzahl möglicher FBS's wird aus CISVARI+E2 entnommen (max. 999).

Beschreibung des Bereichs:

Bytes	Länge	Inhalt
0	4	Gesamtlänge
4	4	Letzte CISUTM - TSN
8	n*66	n Einträge
	4	Endekriterium (High - Value)

Beschreibung eines Eintrags:

Bytes	Länge	Inhalt
0	4	Alte FCB - Adresse
4	4	Neue FCB - Adresse oder noch Nil.
8	4	Kennungen: 10 - OPEN - Typ aus ID1OPEN 11 - Kennung H = HD. V = VD 12 - Kennung E = explizit. I = implizit 13 - SHARUPD - Kennung aus ID1SUSNT
12	54	Dateiname

### 1.3.9 Parameter für CISUTM

Die folgenden Parameter für CISUTM steuern die Größe der Tabellen. Ist CISUTM im Programm CISDBH eingebunden, so wird über die DBH - Parameter gesteuert, ist CISUTM im UTM - Task eingebunden, so müssen die angegebenen Felder in CISVARI "gereppt" werden.

Parameter	UTM - Betrieb Feld	DBH - Betrieb Parameter	Default - Wert
Anzahl Einträge im Speicher	CISVARI + 29C Halbwort	DNRIO =	50
Anzahl Einträge insgesamt *)	CISVARI + 29E Halbwort	DNTRANS =	200
Anzahl Page für MEMDAM (Datenbeschreibungen)	CISVARI + 2A0 Halbwort	DMEMPAGE =	64
Anzahl Pages für ZPV	CISVARI + 2A2 Halbwort	Keiner	64
Anfangsadresse Memory - Pool	CISVARI + 2A8 Wort		0
Monotaskbetrieb?	CISVARI + 2BA Byte		C'N'

\*) So viele Transaktionen können maximal verwaltet werden. Die ersten Bereiche werden im Speicher geführt, die restlichen in der PAM - Datei.

## 1.4 Pool in CISKURZ

Die ersten zwei Worte des Pools dienen der allgemeinen Verwaltung. Das erste Wort (Byte 0 - 3) wird von jedem Teilhaber/ -nehmer mit dem Befehl CS gesetzt (0 oder 1), wenn er den Pool benutzen will. Kann er nicht ran, verliert er mit VPASS 0 die Kontrolle und versucht es immer wieder.

Das zweite Wort enthält die Adresse des nächsten möglichen freien Eintrags. Im leeren Pool enthält dieses Wort die erste mögliche Satzadresse = 00000008.

Die einzelnen Einträge im Pool haben folgenden allgemeinen Aufbau:

4 Bytes Vorwärtspointer	4 Bytes Rückwärtspointer	8 Bytes Transaktion	8 Bytes Kukoschlüssel	2 Bytes Satzkennung
Distanz zum nächsten Satz =0 im letzten.	Distanz zum vorigen Satz = 0 im ersten	TIAM: 8 * x'40'	no = 8 * x'01' %STORE: Speichererkennung	

Mögliche Satzkennungen und zugehörige Satzinhalte sind:

CM	lng(2)	CM + VM	Text	(84)
AZ	"	AZI	Binärzahl	(4)
EB	"	Satz für Passiv - CMD.		( ≤ 32 KB)
LB	"	Satzlänge (LEB) für Passiv.		(4)
KF	"	Fortsetzungsschlüssel wenn ; - Kommando aus KUKO gerufen.		(8)
KK	"	Einzelatz wenn Kommandokette mit ; eingegeben.		(n)
KO	"	Aufbereitetes / substituiertes Kommando.		(n)
KY	"	Folgeschlüssel Terminator / (N)ext GET erzeugt.		(8)
GE	"	Länge (2) Filler (1) Kennung p/c (1) Daten.		(n)
ST	"	Länge (2) Filler (1) Kennung p/c (1) Daten.		(n)
AM	"	Zähler für AM=\$ im Satz mit zugehörigem Key.		(2)
FM	"	Zähler für FM=\$ im Satz mit zugehörigem Key.		(2)
PA	"	Parameterstring aus Aufruf mit neutralem Key.		(n)
PA	"	Parameterstring mit Satzzuordnung nach Substitution.		(n)
RT	"	Key aus GOSUB - CMD für Return.		(8)

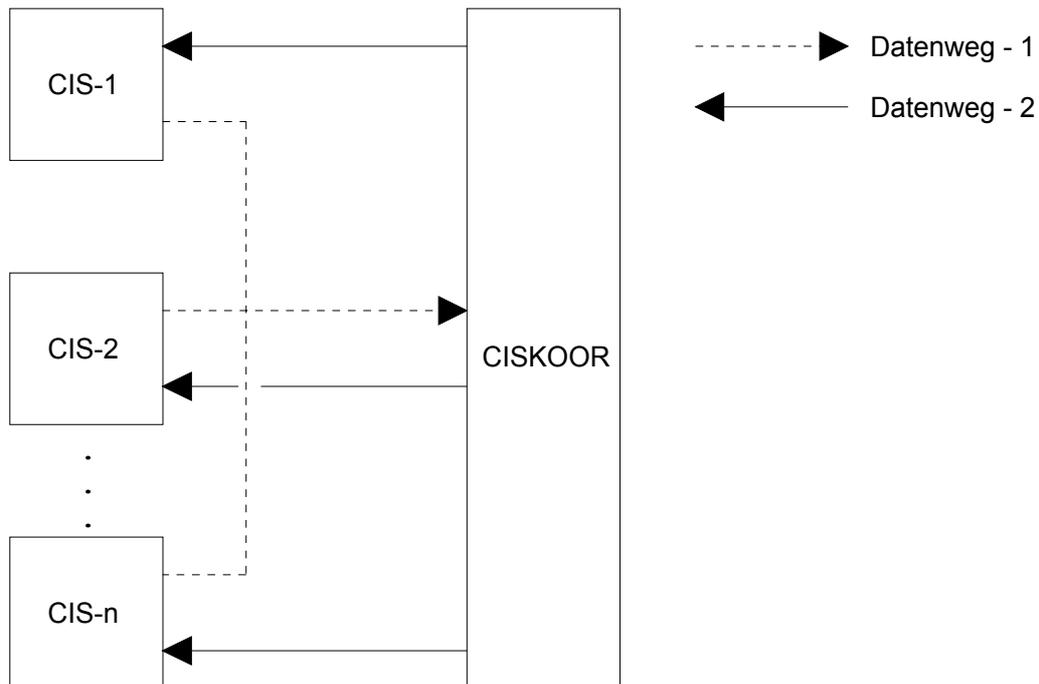
Die Größe des Pools ist 64 KB (0010) und kann per REP in CISKURZ + X'04EE' z.B. auf 256 KB (0040) hochgesetzt werden.



## 2 Kommunikation

### 2.1 Kommunikation CIS - CISKOOR

#### 2.1.1 Übersicht



DV - technisch ist die Kommunikation auf 2 Arten realisiert:

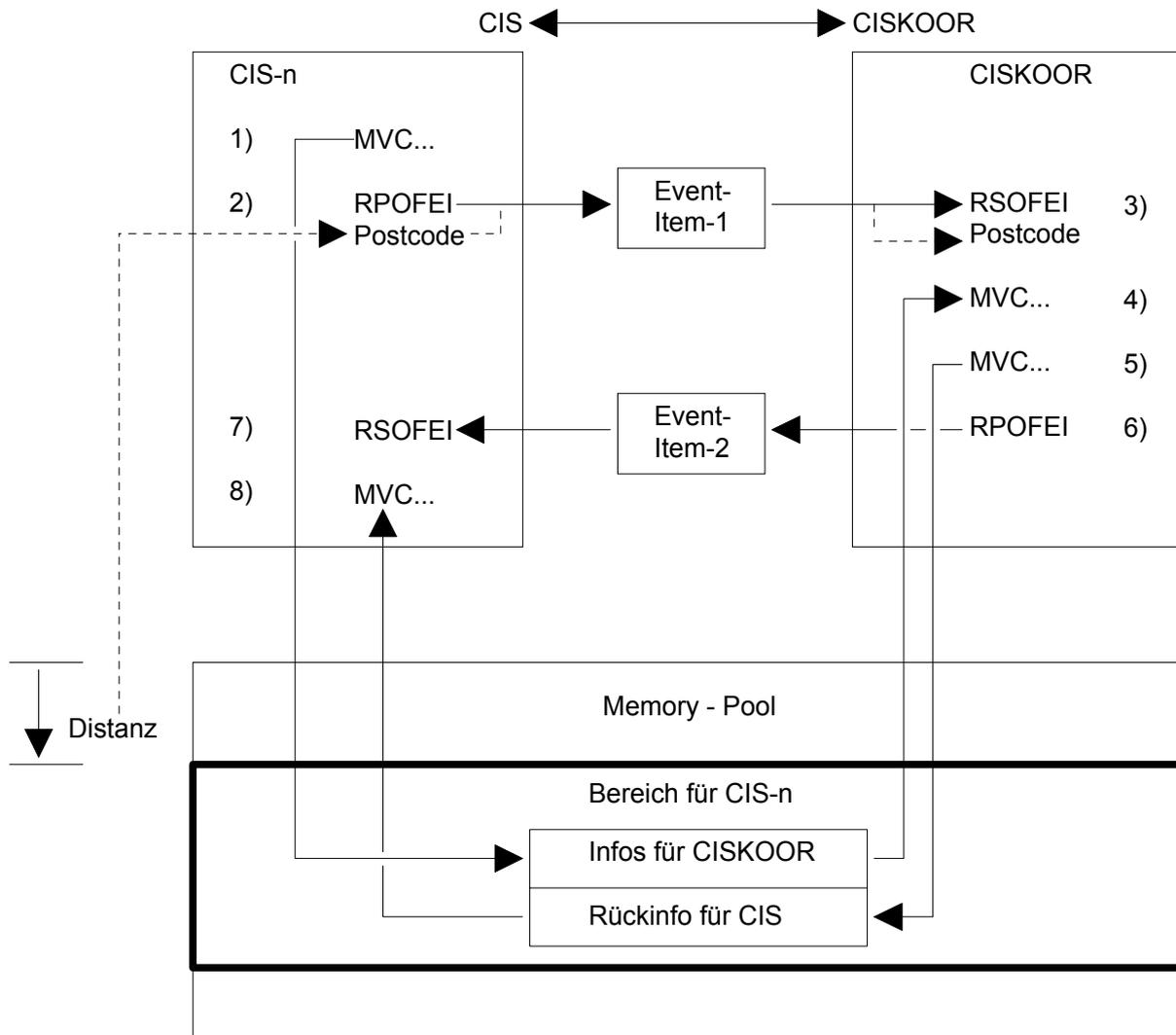
1. Über P1 - Eventing

Nachrichten werden mit RPOFEI angekündigt und mit RSOFEI abgeholt.

2. Über Warteschlange

Nachrichten werden über eine Warteschlange im Memory Pool angekündigt bzw. abgeholt.

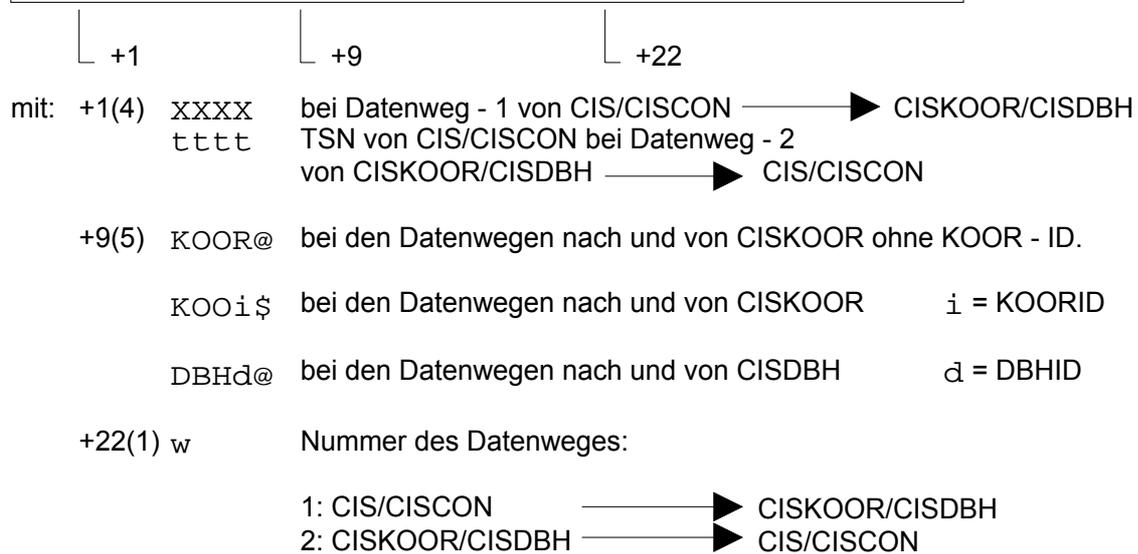
### 2.1.2 DV - technische Realisierung über P1 - Eventing



Erläuterungen:

1. CIS schreibt die Werte für CISKOOR in den ihm zugewiesenen Teil des Memory - Pools.
2. CIS benachrichtigt CISKOOR über das Event - Item 1. Als Postcode wird übermittelt, wo die Informationen im Pool stehen. Da der Pool in jedem Task eine andere Anfangsadresse haben kann, wird hier die Distanz zum Poolanfang angegeben. (Lifetime von RPOFEI = 10 Minuten).
3. CISKOOR empfängt die Nachricht.
4. CISKOOR holt sich alle Informationen aus dem Pool und bearbeitet sie.
5. CISKOOR schreibt die Rückinformation für CIS in den Pool.
6. CISKOOR benachrichtigt CIS über das Event - Item 2. Das Event - Item hat als Namen einen festen Teil und die TSN des CIS - Tasks.
7. CIS empfängt die Nachricht.
8. CIS holt sich alle Rückinformationen aus dem Pool.

### 2.1.3 Namen der Datenwege (Event - Items):



### 2.1.4 Aufbau der Kommunikation

CIS meldet sich beim Memory - Pool an und belegt den benötigten Platz. Dann meldet es sich bei den beiden Event - Items an (hin und zurück).

Eine spezielle Nachricht wird an CISKOOR gesendet. Sie enthält unter anderem die eigene TSN.

CISKOOR nimmt die TSN zur Kenntnis und meldet sich auch bei dem Event - Item für den Rückweg an. (Hat immer die TSN des CIS - Tasks im Namen).

### 2.1.5 Kommunikation über P1 - Eventing

Diese läuft ab, wenn CISKOOR einige Zeit nicht beschäftigt ist und somit auf eine Nachricht von CIS wartet (RSOFEI). Desweiteren läuft sie immer ab, wenn im Poolanfang in den Zellen POOLAKV1 und POOLAKV2 Spaces stehen (siehe Makro POOLA). Dies erreicht man durch den CISKOOR - Parameter KCOMMP1.

### 2.1.5.1 Prinzip

Das P1 - Eventing wird benutzt, um Signale in beiden Richtungen zu senden. Folgende Makros kommen zum Einsatz:

ENAEI zum Anmelden der Event - Items. Ein Item pro Task.  
DPOFEI zum Erzeugen eines RPOFEI - Eintrags.  
DSOFEI zum Erzeugen eines RSOFEI - Eintrags.  
RPOFEI zum Senden eines Signals.  
RSOFEI zum Empfangen eines Signals.

Ein Common - Memory - Pool wird benutzt, um die Daten hin und her zu schicken. Der Anfang des Pools wird von CISKOOR benutzt. Jeder CIS - Task belegt einen ihm ausschließlich zugeteilten Bereich, über den er Daten mit dem Partner in beiden Richtungen austauscht.

### 2.1.6 Kommunikation über Warteschlange

#### 2.1.6.1 Prinzip

Wenn CISKOOR noch eine CIS - Anforderung bearbeitet, und schon eine weitere Anforderung ansteht, dann kann diese in eine Warteschlange eingereiht werden. Kommt CISKOOR ans Ende der Bearbeitung, so prüft er, ob die Warteschlange weitere Anforderungen enthält. Ist dies der Fall, so werden sie bearbeitet. Ist die Warteschlange leer, so wird gewartet, bis wieder eine Anforderung von einem CIS - Task eintrifft (mit RSOFEI). Der CIS - Task, der eine erste Anforderung in eine Warteschlange einreicht, setzt anschließend einen RPOFEI ab, damit CISKOOR wieder geweckt wird.

Bei der Rückmeldung von CISKOOR an CIS, geht CIS, wenn es auf eine Antwort wartet, auf RSOFEI bevor CISKOOR die CIS - Anforderung bearbeitet hat. Ist CISKOOR vor CIS fertig, so kann CIS gleich die Rückmeldung (im Feld POOLR1) auswerten.

#### 2.1.6.2 Elemente der Warteschlange

Die Warteschlange steht in den ersten 4K des Common - Memory - Pools (siehe Makro POOLA). Sie besteht aus folgenden Elementen:

1. POOLAKV1 (+X'74')  
Kennung, damit die Kommunikation von CIS nach CISKOOR über Warteschlange geht. Inhalt: WS oder Spaces.
2. POOLAKV2 (+X'76')  
Kennung, damit die Rückmeldung von CISKOOR nach CIS mit CS - Befehl laufen soll. Inhalt: CS oder Spaces.
3. POOLAWAE (+X'78')  
Anzahl Einträge in der Warteschlange. Zur Zeit 500 = X'000001F4'.
4. POOLAWZ (+X'6C')  
WS - Zähler. CIS erhöht den Zähler bei jedem Senden um 1. CISKOOR verringert den Zähler bei jedem Bearbeiten eines Eintrags um 1.

5. POOLAWZI (+X'70')  
Initialwert für POOLAWZ. (Es ist die Anzahl Tasks, die für den Partner arbeiten; bei Monotask immer 1).
6. POOLAWPC (+X'80')  
CIS - Positionsnummer in der Warteschlange. Jedesmal wenn CIS einen Eintrag füllt, wird der Wert um 1 weitergeschaltet. Anfangswert ist 0. Die Position in der Warteschlange errechnet sich als POOLAWPC Modulo POOLAWZ.
7. POOLAWPK (+X'84')  
CISKOOR - Positionsnummer in der Warteschlange. Jedesmal wenn die Information aus dem aktuellen Eintrag gelesen wurde, wird der Wert um 1 weitergeschaltet. Anfangswert ist 0. Die Position in der Warteschlange errechnet sich als POOLAWPK Modulo POOLAWZ.
8. POOLAWS (+X'88' bis +X'857')  
Warteschlange: 500 Einträge à 4 Bytes. Alle Einträge sind mit Nil initialisiert. Wenn CISKOOR einen Eintrag gelesen hat, wird er mit Nil überschrieben. Die Information, die im Eintrag steht, ist die Distanz von dem für den entsprechenden CIS - Task im Pool zugeteilten Bereich zum Poolanfang. Die Warteschlange wird zyklisch verarbeitet. Damit es keine "Überholeffekte" gibt, löscht CISKOOR jeweils den Eintrag, der 100 Positionen zurückliegt.

### 2.1.6.3 Errechnen der Adresse in der Warteschlange

Die Positionsnummern in POOLAWPC und POOLAWPK werden fortgeschaltet. Da es jeweils ein Wort ist, können die Werte maximal X'FFFFFFF' sein, oder ca.  $4 \times 10^9$ . Wenn jede Sekunde eine Kommunikation stattfindet, dann reichen die Zähler ca. 126 Jahre.

Jeder Postionsnummer entspricht eine Speicherstelle à 4 Bytes (1 Wort). Es ist jedoch nicht für jeden möglichen Wert der Positionsnummer jeweils eine Speicherstelle vorhanden. Es gibt 500 Speicherstellen in der Warteschlange. Diese werden zyklisch bearbeitet, so daß man sich die Bearbeitung auch linear vorstellen kann. Aus der Positionsnummer wird die Adresse wie folgt errechnet:

für CIS:  $(\text{POOLAWPC Modulo POOLAWAE}) * 4 + \text{Adr. POOLAWS} = \text{Adr. Eintrag}$

für CISKOOR:  $(\text{POOLAWPK Modulo POOLAWAE}) * 4 + \text{Adr. POOLAWS} = \text{Adr. Eintrag}$

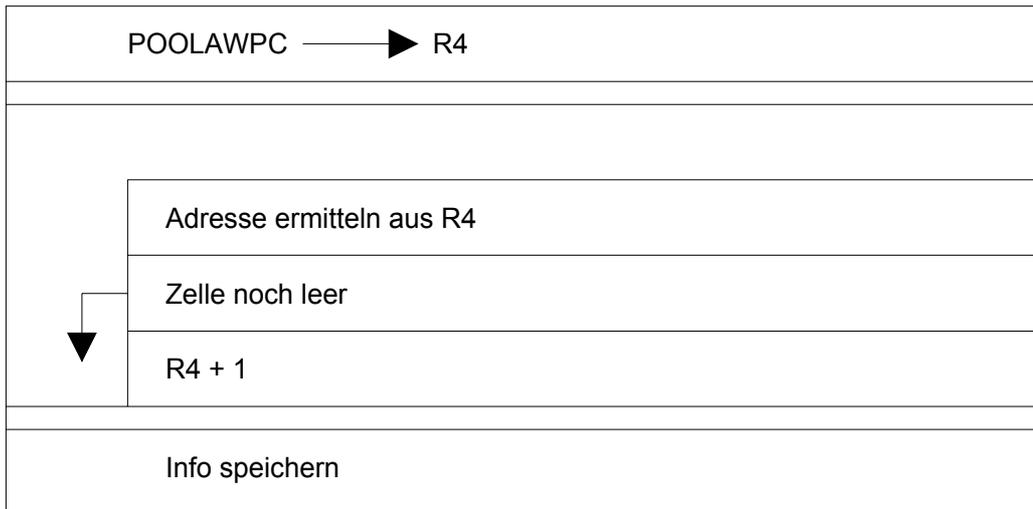
### 2.1.6.4 Eintragen einer Information in die Warteschlange in CIS

In der Routine SEND im Modul COMM1 trägt jeder CIS - Task seine Information in die Warteschlange ein. Die Information, die CIS über die Warteschlange an CISKOOR sendet, ist die Distanz zum Poolanfang des Teils des Pools, der dem CIS - Task zugeordnet ist (Postcode).

Folgende Aktionen laufen ab:

1. Eintrag in Warteschlange belegen (es können mehrere parallele Tasks diese Funktion gleichzeitig ausführen wollen).

Ab dem aktuellen Wert von POOLAWPC wird eine leere Zelle gesucht. (serialisiert mit CS - Befehl). Wird sie nicht gleich gefunden, so wird einfach weitergeschaltet. Es kann sein, daß POOLAWPC nicht auf den nächsten freien Eintrag verweist, sondern etwas davor (siehe nächsten Punkt).



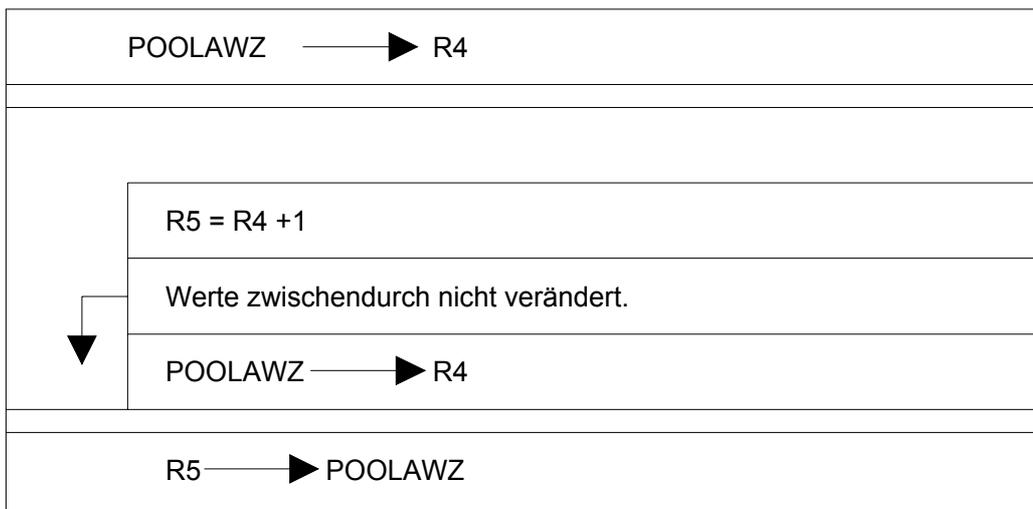
Bemerkung: Die Warteschlange wird zyklisch beschrieben: Nach dem letzten Eintrag wird wieder der erste Eintrag beschrieben.

2. POOLAWPC updaten.

In POOLAWPC soll der Wert gespeichert werden, der der Zelle hinter der eben beschriebenen entspricht. (Vorhergehender Wert von R4 plus 1). Es kann vorkommen, daß ein anderer CIS - Task schon "weiter" ist und schon POOLAWPC geändert hat. Der Wert des aktuellen Tasks wird trotzdem gespeichert. Damit hat zwar POOLAWPC nicht den richtigen Wert, doch durch das Weiterschalten (siehe vorhergehenden Punkt) findet der nächste CIS - Task die nächste freie Zelle.

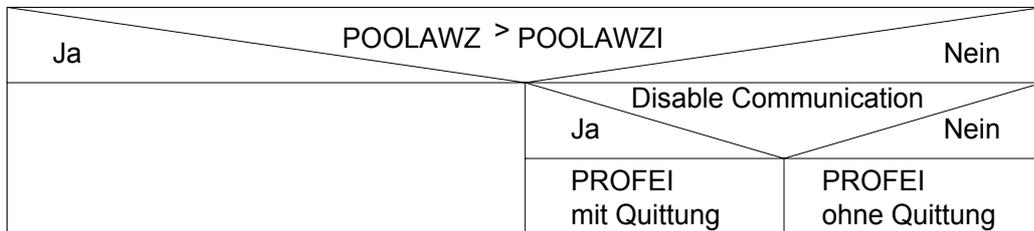
3. POOLAWZ updaten.

Der WS - Zähler POOLAWZ soll immer die Anzahl der Einträge in der Warteschlange enthalten. Er muß also um 1 erhöht werden. Die Logik mit dem CS - Befehl ist nur eine technische Notwendigkeit um sicherzustellen, daß auch 1 auf den aktuellen Wert addiert wird.



4. Eventuell CISKOOR "wecken"

Wenn CISKOOR auf Warten (RSOFEI) steht, weil kein Eintrag mehr in der Warteschlange war, dann muß er mit RPOFEI geweckt werden. Um dies festzustellen wird POOLAWZ mit POOLAWZI verglichen. Ist es größer, dann arbeitet CISKOOR noch. (POOLAWZI ist die Anzahl Tasks, die für CISKOOR arbeiten)



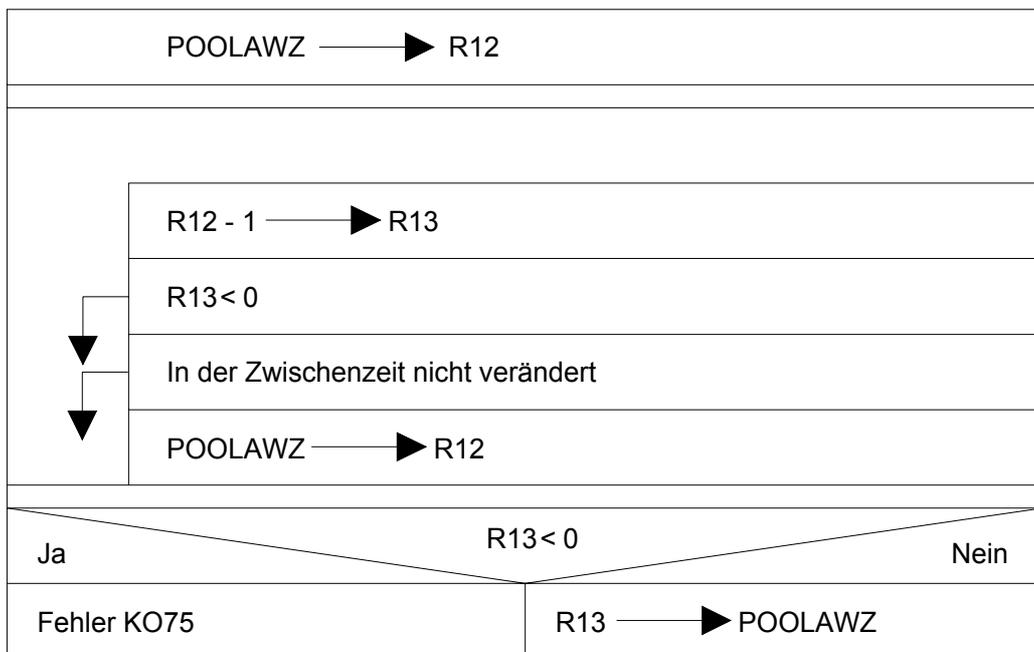
**2.1.6.5 Austragen einer Information aus der Warteschlange in CISKOOR**

In der Routine RCOM liest CISKOOR die Informationen aus der Warteschlange.

Folgende Aktionen laufen ab:

1. POOLAWZ updaten

Der WS - Zähler POOLAWZ wird um 1 zu verringert, da ein Eintrag bearbeitet werden soll. Da beim Einrichten des Memory - Pools POOLAWZ und POOLAWZI mit der Anzahl der TSNs initialisiert werden, darf POOLAWZ nie negativ werden (kann Null werden wenn alle Tasks gleichzeitig arbeiten).



2. Eventuell Warten auf CIS

Wenn POOLAWZ vor dem Herunterzählen nicht größer als POOLAWZI war, dann muß CISKOOR warten, da kein Eintrag in der Warteschlange ist.

3. Info aus Warteschlange nehmen.

POOLAWPK wird um 1 erhöht (serialisiert per CDS - Befehl).

Der gespeicherte Wert minus 1 wird genommen, um die Adresse des Eintrags zu errechnen:

$(\text{POOLAWPK Modulo POOLAWZ}) * 4 + \text{Adr. POOLAWS} = \text{Adr. Eintrag}$

4. Löschen der Info

Geschieht in der Routine SCOM. Es wird jedoch die eben verarbeitete Zelle nicht gelöscht. Es werden immer 100 Zellen verspätet gelöscht, damit es keine "Überholeffekte" seitens CIS geben kann.

### 2.1.6.6 Beschreibung der Rückinformation

Diese Information steht im Feld POOLR1 des Poolsatzes des betreffenden CIS - Tasks:

Folgende Werte gibt es:

1. POOLR1NN (X='00')

wird von CIS vor dem Senden an CISKOOR gesetzt. Dies bedeutet, daß noch nichts abgeschlossen ist. CIS und CISKOOR sind noch aktiv oder werden erst aktiv.

2. POOLR1WA (=X'C0')

wird von CIS gesetzt wenn CIS auf eine Quittung von CISKOOR wartet und noch POOLR1NN im Feld steht (mit CS - Befehl). Dies bedeutet, daß CISKOOR noch aktiv ist und daß CIS auf eine Antwort von CISKOOR wartet. CIS geht auf RSOFEI, CISKOOR auf RPOFEI wenn er fertig ist.

3. Sonstige Werte

werden von CISKOOR gesetzt und informieren über den Verlauf der Bearbeitung.

POOLR1OK (=X'FF') bedeutet, daß alles korrekt verarbeitet wurde. Die anderen Werte sind Fehlernummern. Beim Setzen dieser Werte überschreibt CISKOOR die Werte POOLR1NN oder POOLR1WA.

### 2.1.6.7 Senden der Rückmeldung in CISKOOR

Dies geschieht in der Routine SCOM:

1. Wenn POOLR1 noch den Wert POOLR1NN enthält wird der Rückmeldecode in POOLR1 gespeichert (dies geschieht mit CS - Befehl).
2. Wenn POOLR1 nicht den Wert POOLR1NN enthält, dann ist CIS fertig und wartet mit RSOFEI. Der Rückmeldecode wird in POOLR1 gespeichert und CISKOOR führt einen RPOFEI aus.

### 2.1.6.8 Empfangen der Rückmeldung in CIS

Dies geschieht in der Routine WAIT im Modul COMM1 jedoch nur wenn die Kennung "mit CS - Befehl" gesetzt ist (POOLAKV2=CS). Im anderen Fall wird nur mit RSOFEI (und in CISKOOR mit RPOFEI) gearbeitet.

Folgende Aktionen laufen ab:

1. Wenn POOLR1 noch den Wert POOLR1NN enthält wird POOLR1WA in POOLR1 geschrieben (dies geschieht mit CS - Befehl) und CIS geht auf RSOFEI.
2. Wenn POOLR1 einen anderen Wert enthält wird dieser Wert als Rückmeldung verarbeitet. CIS geht nicht auf RSOFEI.

### 2.1.6.9 Übersicht der Werte in POOLR1

Wert	Zustand von	
	CIS	CISKOOR
POOLR1NN = X'00'	aktiv	aktiv
POOLR1WA = X'C0'	wartet	aktiv
Sonstige	?	fertig

### 2.1.7 Abbau der Kommunikation

Das eigentliche Senden geschieht wie bei der Kommunikation beschrieben. Der Wert der übermittelt wird (COSPOST) ist:

X'FF00iiii'

wobei iiii die Nummer (id.) von CIS in den Tabellen von CISKLOOR ist.

CISKLOOR findet an Hand der Nummer den Eintrag in seiner Tabelle.

Der Platz im Memory - Pool wird von CISKLOOR freigegeben.

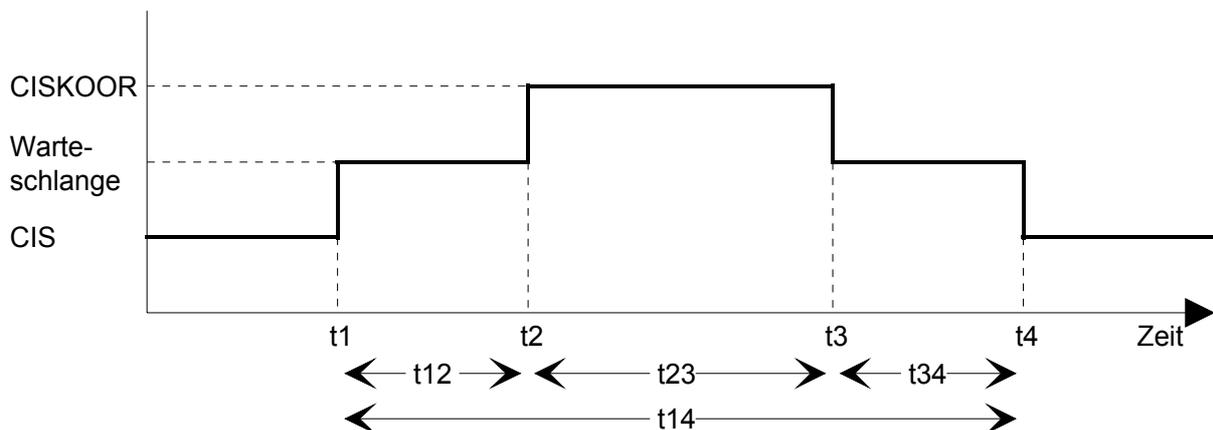
Ist CISKLOOR mit RPOFEI aufgerufen worden, so war es ein Senden mit Quittung. CIS wartet bis die Quittung eingetroffen ist und meldet sich nach Freigabe der beiden Event - Items vom Memory - Pool ab.

### 2.1.8 Zeitmessungen

Um etwas über die Dauer der Kommunikation zwischen CIS und CISKLOOR zu erfahren, wird an einigen Punkten die Zeit gemessen (per Befehl STCK. Schnell da kein SVC). Die Zeitdifferenzen werden gebildet und in CISKLOOR aufaddiert. Am Ende des CISKLOOR - Laufes werden diese Werte mit anderen Statistiken ausgegeben. (sofern angefordert mit dem KSTAT - Parameter).

Interessante Werte sind die Wartezeiten in den Warteschlangen hin und zurück, sowie die Verweilzeit in CISKLOOR.

Die folgende Grafik veranschaulicht diese Punkte:



Diese Zeitmessungen entfallen bei inlinked CISKLOOR.

Die Zeiten werden immer für einen kompletten Zyklus gemessen und dann im Poolsatz an CISKLOOR übergeben. Somit fehlt in der Messung jeweils die erste (ENABLE) und letzte (DISABLE) Messung.

## 2.2 Kommunikation CISCON - CISDBH

### 2.2.1 Allgemeines

Es gibt zwei Möglichkeiten der Kommunikation zwischen CISCON und CISDBH:

1. P1 - Eventing und Memory Pool

Ablauf wie bei der Kommunikation zwischen CIS und CISKOOR.

2. DCAM

Es werden über DCAM Sätze mit V - Format ausgetauscht. Hierzu wird die CALL - Schnittstelle von ihrer parallelen Struktur in eine serielle Struktur umgewandelt. CISCON analysiert das jeweilige CIS - Kommando und baut je nach Kommandotyp einen String auf.

### 2.2.2 Kommunikation über P1 - Eventing

Läuft ab wie die Kommunikation von CIS zu CISKOOR. Siehe Beschreibung unter CISKOOR (Vgl. Seite 35). Gegebenenfalls ist CISKOOR durch CISDBH und CIS durch CISCON zu ersetzen.

### 2.2.3 Kommunikation über DCAM

#### 2.2.3.1 Allgemeines

Beide Partner CISCON und CISDBH sind als DCAM - Anwendungen miteinander verbunden. Die Daten werden als variabel lange Sätze untereinander ausgetauscht.

#### 2.2.3.2 Aufbau der Kommunikation

Jeder Partner meldet sich bei DCAM als Anwendung an. Die Namen sind:

CISDBHi = Name der DBH - Anwendung (i = DBHID)

CISStttt = Name der CISCON - Anwendung (tttt = TSN von CISCON)

CISCON baut die Verbindung zu CISDBH auf. Dieser hat eine LOGON - Contingency, die den Aufbau annimmt wenn die Verbindungsnachricht stimmt:

```
X'00'  
C'CIS-C10-'  
X'112233'
```

In allen anderen Fällen wird mit YREJLOG abgebrochen.

## Kommunikation

Der Wert von MAXLN (maximale Länge einer Nachricht auf dieser Verbindung) wird gespeichert. Allerdings darf eine Nachricht nicht länger als KLEN (4096 Bytes) sein, auch wenn MAXLN größer ist.

An CISDBH wird eine Nachricht mit dem Code "Enable" gesendet und eine Rückmeldung erwartet. Die Nachricht enthält TSN, Betriebsart, Applinamen und Hostnamen von CISCON. Somit kennt CISDBH seinen Partner.

Der DB - Status wird auf "Connected" gesetzt. Hat CISDBH eine Paßwortliste (Parameter DPASS), so wird sie bei der Rückmeldung übermittelt.

### **2.2.3.3 Datenverkehr**

#### **2.2.3.3.1 Senden an CISDBH**

Der CIS - Call wird "serialisiert". Dazu müssen alle CIS - Kommandos dem Connection - Modul bekannt sein.

Der daraus resultierende String wird mit YSEND an CISDBH gesendet.

#### **2.2.3.3.2 Empfangen der Antwort**

Wird ergänzt

#### **2.2.3.4 Abbau der Kommunikation**

Wird ergänzt

## 2.2.4 Zeitmessungen

Um etwas über die Dauer der Kommunikation zwischen CISCON und CISDBH zu erfahren, wird an einigen Punkten die Zeit gemessen (Mit Befehl STCK. Schnell da kein SVC). Die Zeitdifferenzen werden gebildet und in CISDBH aufaddiert. Am Ende des CISDBH - Laues werden diese Werte mit anderen Statistiken ausgegeben. (sofern angefordert per DSTAT - Parameter).

### 2.2.4.1 Kommunikation über P1 - Eventing

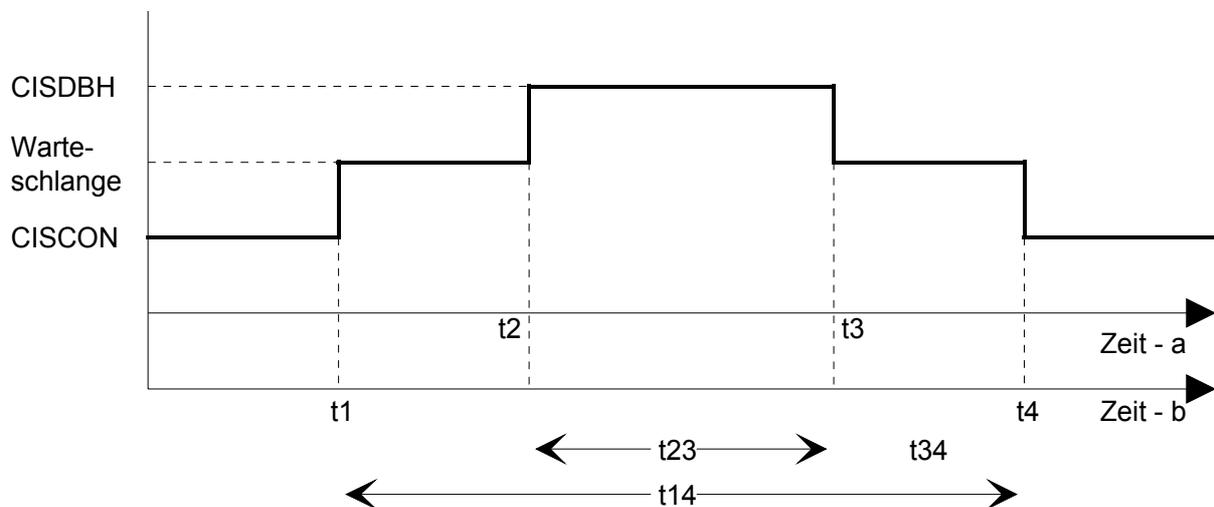
Die Messungen werden analog zur Kommunikation CIS - CISKOOR (vgl. Seite 44) ausgeführt.

### 2.2.4.2 Kommunikation über DCAM

Es ist zu beachten, daß die Zeit in CISDBH eine absolut andere ist als die Zeit in CISCON (verschiedene Rechner, da Kommunikation über DCAM). Es können also keine Differenzen mit diesen verschiedenen Zeitachsen gebildet werden.

Interessante Werte sind die Wartezeiten in den Warteschlangen hin und zurück, sowie die Verweilzeit in CISDBH. Da die Wartezeiten nicht direkt erfaßt werden können (siehe oben) wird die Gesamtzeit  $t_{14}$  erfaßt und beim nächsten Aufruf an CISDBH gemeldet. (Es fehlen die Werte für den ersten und den letzten Aufruf).

Die folgende Grafik veranschaulicht die Meßpunkte:



### 2.2.5 Beschreibung des DBS - Feldes

Das Feld DBS ("DB - Status") wird benötigt wenn mit UTM im synchronisierten Betrieb gearbeitet wird. In diesem Feld werden Statusinformationen zwischen CISCON und CISDBH ausgetauscht.

Das Feld ist ein Byte groß. Die Werte haben folgende Bedeutungen:

Name	Wert	Bedeutung
DBSCONDB	1	DBH - Connection ausgeführt.
DBSDISB	2	DBH - Disconnction ausgeführt.
DBSDWNDB	3	DBH ist "down".
DBSCONTA	4	Transaktion wurde begonnen.
DBSDISTA	5	Transaktion wurde beendet.
DBSRSTTA	6	Transaktion wurde zurückgesetzt.
DBSCLITA	7	Ein Transaktionsende wurde vom Benutzer verlangt.
DBSUPDTA	8	Die Transaktion verändert Daten.
DBSTAUNK	9	Transaktion unbekannt (Statusabfrage).
DBSTARST	10	Transaktion zurückgesetzt (Statusabfrage).
DBSCONTR	11	Lesetransaktion wurde begonnen.
DBSUCALL	225	Der Aufruf kommt vom Benutzer und nicht vom DC - System.

## Setzen der Werte im Feld DBS

Inhalt DBS		Wird gesetzt in			Bemerkung
Wert	Name	Programm	Modul	Routine	
0		CISCON	CISCO1	DB...	Init. nicht user - call
255	DBSUCALL	CISCON	CISCO1	DB...	Init. user - call
1	DBSCONDB	CISCON	COMM1	ENAI	Wenn CODBHN $\neq$ KOOR und erfolgreicher Verbindungsaufbau.
			DCAM	ENAI	Bei erfolgreichem Verbindungsaufbau.
2	DBSDISDB	CISCON	COMM1	DIS	Wenn CODBHN $\neq$ KOOR und erfolgreicher Verbindungsaufbau.
			?	?	Bei erfolgreichem Verbindungsaufbau.
3	DBSDWNDB	CISCON	COMM1	SEND	Wenn CODBHN $\neq$ KOOR und CO09 oder CO27
			COMM1	WAIT	
			DCAM	SEND	Wenn DC14/DC15/DC16
			DCAM	WAIT	
4	DBSCONTA	CISDBH	COMM2	CHKPR	TR,A
5	DBSDISTA	CISDBH	COMM2	CHKPR	TR,E
6	DBSRSTTA	CISDBH	COMM2	CHKPR	Reset TID Reset TSN
		CISCON	CISCO1	DBADI	Rücksetzen DBSRSTTA
7	DBSCLITA	CISDBH	ICHKPT	END	Nur wenn DBSUCALL gesetzt ist 1)
8	DBSUPDTA				2)
9	DBSTAUNK	CISDBH	ICHKPT	STA	Wenn RUECK = F'1' (aus CISBIM)
10	DBSTARST	CISDBH	ICHKPT	STA	Wenn RUECK = F'2' (aus CISBIM)
11	DBSCONTR	CISDBH	COMM2	CHKPR	TR,AL

1) CISDBH wird nicht (mehr) aufgerufen. DBSCLITA wird nicht benötigt. In CISCON wird beim User - Call mit TR,E für UTM TAMTACLI gesetzt.

2) In CISCON wird automatisch bei DBSCONTA wie bei DBSUPDTA verfahren: Setzen von TAMTAUPD.

Rückmeldungen an UTM

	Inhalt von DBS		Setzen TSKMSTAT	Setzen/Rücksetzen TAMSDBTA		Setzen TAMRBR	Setzen TAMUTMCO
	Name	Setzen DBCERRCD Defaultwert= DBCOK		S/R 1)	Wert		
1	DBSCONDB		TSKMCONC				
2	DBSDISDB		TSKMDCON				
3	DBSDWNDB	DBCDBDWN 4)	TSKMADCO				
4	DBSCONTA			S S	TAMTAOPN TAMTAUPD 3)		TAMMULTI TAMUCLRQ
5	DBSDISTA			S R	TAMTACL TAMTACAN		
6	DBSRSTTA	DBCTARES 2)		S R R R	TAMTACAN TAMTAOPN TAMTACLI TAMTACL	TAMRBR 2)	
7	DBSCLITA			S R	TAMTACLI TAMTACAN		
8	DBSUPDTA			S	TAMTAUPD		
9	DBSTAUNK			R	alles		
10	DBSTARST			S	TAMTACAN		
11	DBSCONTR			S	TAMTAOPN		TAMMULTI TAMUCLRQ

- 1) S=Setzen, R=Rücksetzen
- 2) Nur bei User - Call
- 3) Durch Ansprung DBSUPDTA - Behandlung
- 4) Zusätzlich UTM - Error message

### 3 Spezielle Dateien

#### 3.1 Datensicherungsdateien

##### 3.1.1 Allgemeines

Es gibt im Prinzip zwei Datensicherungsdateien: After - Image Datei  
Before - Image Datei

Die Protokolldatei wird hier beschrieben, weil sie von CISKLOOR erstellt wird.

Die Statusdatei wird hier beschrieben, weil sie in der Before - Image Datei gespeichert ist.

Die Dateien der Datensicherung werden mit dem Programm CISINIT initialisiert und vom Programm CISKLOOR geschrieben. Der Inhalt wird mit den Programmen CISKLOOR (Before - Image Restart), CISRA (After - Image Recovery) und CISPROT (Ausdruck der After - Image Datei und der Protokolldatei) ausgewertet.

Alle Datensicherungsdateien sind PAM - Dateien, die Protokolldatei ist eine ISAM - Datei.

Die erste Seite(Page) enthält Verwaltungsinformationen, ab der zweiten Seite stehen die Daten.

Die PAM - Schlüssel werden seit CIS V8.0 nicht mehr benützt.

Aufbau einer PAM - Page in der AIM - Datei:

	File - ID	Res. -0	Version	Distanz	Anz. Daten	Daten	Res.
Länge:	4	4	4	2	2	2032	0
Distanz:	0	4	8	12	14	16	
<p>Länge PAM - Page (2K = 2048)</p>							

File - ID: wird bei jedem CISINIT - Lauf neu vergeben.  
(Sekunden seit 1.1.1985 - eindeutig bis 31.12.2035).

## Spezielle Dateien

Aufbau einer PAM - Page in der BIM - Datei:

	File - ID	Res	Ver.	P - R	P - V	K.	R.	TID	R.	Dist.	Anz. Dat.	Daten	Res
Länge:	4	4	4	4	4	1	1	2	4	2	2	2016	0
Distanz:	0	4	8	12	16	20	21	22	24	28	30	32	

← Länge PAM - Page (2K = 2048) →

Mit den Informationen am Anfang der Page kann festgestellt werden, bis wohin eine Datei beschrieben wurde.

### Bemerkungen zur Dateiversion

In der After - Image - Datei wird bei jedem Initialisieren mit CISINIT die Versionsnummer um 1 erhöht.

In der Before - Image - Datei wird die Versionsnummer beim Initialisieren mit CISINIT auf 1 gesetzt. Bei jedem Transaktionsanfang wird die Versionsnummer um 1 erhöht und in der Page dieser Transaktion benützt. Die bisher höchste Nummer steht in der ersten Seite.

### 3.1.2 Beschreibung der After - Image - Datei

Die erste PAM - Seite enthält diverse Verwaltungsinformationen. Ab der zweiten PAM - Seite stehen die After - Image - Sätze.

#### 3.1.2.1 Beschreibung der ersten PAM - Seite

Die erste PAM - Seite hat folgendes Format:

Bytes	Länge	Format	Inhalt	Woher
0 - 3	4	B	"File - Id"	Init
4 - 7	4		Reserve - 0	
8 - 11	4	B	Version der Datei	Init
12 - 13	2	B	Distanz zu den Daten (16)	Init
14 - 15	2	B	Länge der Daten	Init
16 - 23	8	C	Kennung: #CISAIM#	Init
24 - 27	4	B	Last Page. Nur gültig wenn Zustand C	Init Close
28 - 33	6	C	Datum letzter OPEN (JJNNN)	Init & Open
34 - 39	6	C	Uhrzeit letzter OPEN (HHMMSS)	
40 - 45	6	C	Datum letzter CLOSE (JJNNN)	Init & Close
46 - 51	6	C	Uhrzeit letzter CLOSE (HHMMSS)	
52 - 52	1	C	Kennung I: Initialisiert A: Enthält After - Images	Init & Open
53 - 53	1	C	Zustand O: Output eröffnet C: geschlossen	Open Close
54 - 54	1	C	KOORID nach Init :X'00' nach 1. Open : Id, X'40' = Noid	
55 - 2047	1993		Frei, enthält X'FF'	Init

### 3.1.2.2 Beschreibung eines After - Image - Satzes

Ein After - Image - Satz hat folgendes Format:

Bytes	Länge	Format	Inhalt
0 - 3	4	Binär	Länge des gesamten Satzes. Format: $\ell\ell + 2$ Spaces
4 - 7	4	Binär	Länge des Kopfteils. Format: $\ell\ell + 2$ Spaces
8 - 13	6		Datum Start CISKOOR: JJNNN + 1 Space
14 - 19	6		Uhrzeit Start CISKOOR: HHMMSS
20 - 23	4	Binär	Laufende Nummer seit Start CISKOOR
24 - 24	1	Zeichen	Satztyp: A
25 - 25	1	Zeichen	Zusatz - Typ: C: (Checkpoint) = Transaktion D: Differenz - Satz E: Elim - Satz I: I Insert - Satz S: Store - Satz X: "Extraktionssatz"
26 - 26	1	Zeichen	Bei Zusatz - Typ = C sonst: E: "explizit" = TR,A / TR,E I: "implizit" intern vergeben Space
27 - 27	1	Zeichen	Bei Zusatz - Typ = C sonst: A: TR,A Anfang E: TR,E Ende Space
28 - 28	1	Zeichen	vor CISRA - Lauf: nach CISRA - Lauf: Space O = O.K. } in Transaktion F = Fehler }
29 - 29	1	Zeichen	Betriebsart: D: DBH N: TIAM S: UTM - Synchronisiert U: UTM - Unsynchronisiert
30 - 33	4	Numerisch	TSN des Aufrufers
34 - 41	8	Alpha	Transaktion - Id.      Betrieb = N: TSN + 4 Spaces S: UTM - interne Kennung U: KCLOGTER / KCBENID
42 - 49	8	Alpha	Applikation oder Spaces (Betrieb = N)
50 - 57	8	Alpha	Hostname oder * + 7 Spaces (= eigener Host, bei Betrieb = N)
58 - 111	54	Alpha	Vollqualifizierter Dateiname
112 - 112	1	Binär	RECFORM:      X'04' = F X'02' = V
113 - 113	1	Binär	KEYLEN - 1
114 - 115	2	Binär	KEYPOS - 1
116 - 116	1	Binär	Blocksize
117 - 117	1	Binär	Open - Typ (aus FCB) z.B. x'40' OUTIN x'20' INOUT x'04' OUTPUT
118 - 121	4	Binär	Länge des Datenteils. Format: $\ell\ell + 2$ Spaces
122 -			Datenteil (maximal 32 K). Der Inhalt richtet sich nach dem Zusatztyp

Format bei Zusatz - Typ C (ab Byte 58)

Bytes	Länge	Format	Inhalt
58 - 65	8		Nil
66 - 71	6	Numerisch	Uhrzeit

Format bei Zusatz - Typ = X (ab Byte 122)

Bytes	Länge	Format	Inhalt
122 - 175	54	Alpha	Vollqualifizierter Name der HD
176 - 177	2	Binär	Länge des "Extraktionssatzes"
178 - 178	1		nicht benützt
179 - 179	1	Binär	OP - Code x'00' = Einfügen in VD x'01' = Löschen in VD x'10' = Verdichten x'20' = Neuer EX - Satz ab V11 ab V11
180 - 182	3	Alpha	Segment (Kurzname) bis V10
180 - 183	4	Alpha	Segment ab V11
183 - 183	1	Binär	Länge Ordnungsbegriff (o) bis V10
184 - 184	1	Binär	Länge Ordnungsbegriff (o) ab V11
184 - 183 + o	o		Ordnungsbegriff bis V10
185 - 184 + o	o		Ordnungsbegriff ab V11
184 + o	1	Binär	Länge des Feldinhalts (a) bis V10
185 + o	1	Binär	Länge des Feldinhalts (a) ab V11
185 + o bis 184+o+a	a		Feldinhalt bis V10
186 + o bis 185+o+a	a		Feldinhalt ab V11

Format bei Zusatz - Typ = S/I/E/D (ab Byte 122):

Der Datenteil enthält: bei Zusatz - Typ S / I: Isam - Satz  
 bei Zusatz - Typ D: Isam - Schlüssel  
 bei Zusatz - Typ D: Isam - Schlüssel  
 Differenzinformationen mit folgendem Format:  
 2 Bytes: Position im alten Satz.  
 2 Bytes: Länge alter Inhalt.  
 2 Bytes: Länge neuer Inhalt.  
 x Bytes: neuer Inhalt.

### 3.1.2.3 Beispiel

Das folgende Beispiel soll die Aufteilung der After - Image - Sätze auf die einzelnen PAM - Seiten veranschaulichen.

Die After - Image - Datei enthält z.B. folgende Sätze:

- Satz - 1: 400 Bytes in PAM - Seite 2
- Satz - 2: 100 Bytes in PAM - Seite 2
- Satz - 3: 2.000 Bytes in PAM - Seite 2 + 3
- Satz - 4: 10.000 Bytes in PAM - Seiten 3 bis 8
- Satz - 5: 100 Bytes in PAM - Seite 8

Die Datei hat folgenden Aufbau:

Pam - Seite	Kopf der Seite		Daten		
	Bytes 12 - 13	Bytes 14 - 15			
2	0	2032	Satz - 1 0 - 399	Satz - 2 400 - 499	Satz - 3 (A) 500 - 2031
3	468	2032	Satz - 3 (B) 0 - 467	Satz - 4 (A) 468 - 2031	
4	x'FFFF'	2032	Satz - 4 (B)		
5	x'FFFF'	2032	Satz - 4 (C)		
6	x'FFFF'	2032	Satz - 4 (D)		
7	x'FFFF'	2032	Satz - 4 (E)		
8	308	408	Satz - 4 (F) 0 - 307	Satz - 5 308 - 407	

### 3.1.2.4 Beschreibung eines Checkpoint - Satzes

Die Checkpoint - Sätze stehen in der After - Image - Datei zwischen den eigentlichen After - Image - Sätzen.

Es gibt 2 Arten von Checkpoint - Sätzen

1. "implizite" Sätze
2. "explizite" Sätze

Diese werden in folgenden Fällen geschrieben:

1. Bei TR,A  
wird ein "expliziter" Checkpoint - Anfang - Satz geschrieben.
2. Bei TR,E  
wird ein "expliziter" Checkpoint - Ende - Satz geschrieben.
3. Vor dem Ändern eines HD - Satzes  
wird ein "impliziter" Checkpoint - Anfang - Satz geschrieben. (Nur wenn ohne Transaktionen gearbeitet wird).
4. Nach dem Ändern des letzten zugehörigen VD - Satzes wird ein "impliziter" Checkpoint - Ende - Satz geschrieben. (Nur wenn ohne Transaktionen gearbeitet wird).

Folgendes Beispiel soll die Reihenfolge der Sätze in der After - Image - Datei veranschaulichen:

Es sollen in einer Transaktion 2 verschiedene HD - Sätze geändert werden. Es ergibt sich dann folgende Reihenfolge:

CIS - Kommando	Satz in der After - Image - Datei
TR,A SUCHEN... SP,Z AENDERN...	ECPL. CHECKP. ANFANG  HD - After Image 1 VD - After Image 11 . . . VD - After Image 1n
SUCHEN... SP,Z AENDERN  TR,E	HD - After Image 2 VD - After Image 21 . . . VD - After Image 2n EXPL. CHECKP. ENDE

Bemerkung: Seit CIS V7.02 werden die impliziten Checkpunkte nicht mehr innerhalb einer Transaktion geschrieben.

Spezielle Dateien

Format eines Checkpoint - Satzes:

Siehe Beschreibung des After - Image - Satzes auf Seite 54

Zusatz - Typ = C

### 3.1.3 Beschreibung der Before - Image - Datei

Die erste PAM - Seite enthält diverse Verwaltungsinformationen. Die PAM - Seiten 2 bis n sind für Statussätze reserviert. Danach werden BIM - Sätze geschrieben. Reichen die ersten Pages nicht für Statussätze aus, so werden diese auch zwischen den BIM - Sätzen gespeichert. Wird vorne wieder Platz frei, so werden die Statussätze nach vorne geholt.

Der Wert n wird bei der Initialisierung der BIM - Datei auf 100 gesetzt sofern die Datei mindestens so groß ist. Im andern Fall wird die Dateigröße genommen.

#### 3.1.3.1 Beschreibung der ersten PAM - Seite

Die erste PAM - Seite hat folgendes Format:

Bytes	Länge	Format	Inhalt	Woher
0 - 3	4	B	"File - Id"	Init
4 - 7	4		Reserve - 0	
8 - 11	4	B	Version der Datei	Init
12 - 15	4	B	Pointer - Rück (hier 0)	Init
16 - 19	4	B	Pointer - Vor (hier 0)	Init
20 - 20	1	C	Kennung Satz 6	Init
21 - 27	7		Reserve	
28 - 29	2	B	Distanz zu den Daten (32)	Init
30 - 31	2	B	Länge der Daten	Init
32 - 39	8	C	Kennung: #CISBIM#	Init
40 - 43	4	C	C'OPN:'	Init
44 - 49	6	C	Datum letzter OPEN (JJNNN)	Init &
50 - 55	6	C	Uhrzeit letzter OPEN (HHMMSS)	OPEN
56 - 59	4	C	C'CLS:'	Init
60 - 65	6	C	Datum letzter CLOSE (JJNNN)	Init &
66 - 71	6	C	Uhrzeit letzter CLOSE (HHMMSS)	Close
72 - 75	4	C	C'FLG:'	Init
76 - 76	1	C	Kennung: I: Initialisiert B: Enthält Before - Images	Init & Open
77 - 77	1	C	Kennung - 2: I: Initialisiert S: Enthält Statussätze	Init & Open
78 - 78	1	C	Zustand: O: Output eröffnet C: Geschlossen	Open Close
79 - 79	1	C	KOORID: nach Init: X'00' nach 1. Open: Id, X'40' = NOID	

## Spezielle Dateien

Bytes	Länge	Format	Inhalt	Woher
80 - 83	4	C	C'GEN:'	Init
84 - 87	4	B	Anzahl PAM - Pages pro Segment	Init
88 - 91	4	B	Last Page	Init & CIS
92 - 95	4	C	C'STA:'	Init
96 - 99	4	B	1. Page für Statusätze	Init
100 - 103	4	B	Letzte Page für Statusätze	Init & CIS
104 - 107	4	B	Letzte beschriebene Page für Statusätze	Init & CIS
108 - 111	4	C	C'BIM:'	Init
112 - 115	4	B	Nr. der 1. Page für BIMs	Init
116 - 119	4	C	C'FRE:'	Init
120 - 123	4	B	Anfang Free - list	Init & CIS
124 - 127	4	B	Ende Free - list	Init & CIS

### Erläuterungen:

Ab CIS V11 ist die Verwaltung der BIM - Datei gegenüber den Vorgängerversionen komplett umgestellt worden. Den einzelnen Segmenten sind nicht mehr Bytes im Inhalt zugewiesen. Die Pages, die einer Transaktion zugeordnet sind, werden untereinander mit Vor - Pointern und mit Rück - Pointern verkettet. Der Transaktionseintrag im Speicher verweist auf diese Kette von PAM - Pages.

Bei Transaktionsende wird die gesamte Kette der PAM - Pages vor die Free - List gehängt.

Beim Rücksetzen einer Transaktion werden die einzelnen BIM - Sätze von hinten nach vorne verarbeitet. Die Pointer in den PAM - Pages und im Eintrag im Speicher helfen hierbei.

Werden neue PAM - Pages für eine Transaktion benötigt, so werden sie erst aus der Free - List entnommen, dann wird die Datei erweitert. Das Problem, daß die BIM - Datei nicht mehr wachsen kann weil das Inhaltsverzeichnis voll ist, gibt es nicht mehr.

Die Statusätze stehen auch in der BIM - Datei. Damit die Verwaltung nicht zu kompliziert wird, sind am Anfang der Datei PAM - Pages hierfür reserviert. Wenn möglich, sind die Pages 2 bis 100 reserviert. Ist die Datei beim Init - Lauf kleiner als 100 Pages so wird die Dateigröße genommen.

Reicht dieser Platz nicht für die Statusätze, so wird für sie Platz aus der Free - List entnommen wie für BIM - Sätze. Am Ende eines Laufs wird dann versucht, die Statusätze nach vorne zu schieben.

### 3.1.3.2 Beschreibung eines Before - Image - Satzes

Ein Before - Image - Satz hat folgendes Format:

Bytes	Länge	Format	Inhalt
0 - 3	4	Binär	Länge des gesamten Satzes. Format: <i>ll</i> + 2 Spaces
4 - 7	4	Binär	Länge des Kopfteils. Format: <i>ll</i> + 2 Spaces
8 - 13	6		Datum Start CISKOOR: JJNNN
14 - 19	6		Uhrzeit Start CISKOOR: HHMMSS
20 - 23	4	Binär	Laufende Nummer seit Start CISKOOR
24 - 24	1	Zeichen	Satztyp: B
25 - 25	1	Zeichen	Zusatz - Typ: E: Elim - Satz I: Insert - Satz S: Store - Satz X: "Extraktionssatz"
26 - 28	3		Reserve
29 - 29	1	Zeichen	Betriebsart: D: DBH N: TIAM S: UTM - Synchronisiert U: UTM - Unsynchronisiert
30 - 33	4	Numerisch	TSN des Aufrufers
34 - 41	8	Alpha	Transaktion - Id.      Betrieb = N: TSN + 4 Spaces S: UTM - interne Kennung U: KCLOGTER / KCBENID
42 - 49	8	Alpha	Applikation oder Spaces
50 - 57	8	Alpha	Hostname oder * + 7 Spaces (= eigener Host)
58 - 111	54	Alpha	Vollqualifizierter Dateiname
112 - 112	1	Binär	RECFORM:      X'04' = F X'02' = V
113 - 113	1	Binär	KEYLEN - 1
114 - 115	2	Binär	KEYPOS - 1
116 - 116	1		Reserve
117 - 117	1	Numerisch	Kennung:      C'1': Erstes Mal angesprochen C'2': Alle weiteren Male
118 - 121	4	Binär	Länge des Datenteils. Format: <i>ll</i> + 2 Spaces



### 3.1.4 Beschreibung der Protokolldatei

Die Protokolldatei ist keine Datensicherungsdatei. Sie wird jedoch zentral von CISK00R beschrieben, damit die zeitliche Reihenfolge der Kommandos protokolliert werden kann. Sie ist für Diagnose- und Überwachungsaufgaben gedacht.

Die Protokolldatei ist ab V11 eine ISAM - Datei mit:

```
KEYPOS=9
KEYLEN=16
RECFORM=V
BLKSIZE=(STD,1)
```

#### 3.1.4.1 Beschreibung des ersten Satzes

Der erste Satz der Protokolldatei hat folgenden Aufbau:

Bytes	Länge	Format	Inhalt
0 - 3	4	Hex.	Satzlängenfeld ( <i>ℓℓbb</i> )
4 - 7	4	Hex.	Nils
8 - 23	16	Hex.	ISAM - Key (= Nils)
24 - 31	8	Zeichen	Kennung C'#CISPRO#'
32 - 43	12	Numerisch	Zeitpunkt letzter Open (Datum, Uhrzeit). Format: JJNNN HHMMSS
44 - 55	12	Numerisch	Zeitpunkt letzter Close (Datum, Uhrzeit). Format: JJNNN HHMMSS
56 - 56	1	Zeichen	Kennung I: Datei ist initialisiert. P: Datei enthält Protokollsätze.
57 - 57	1	Zeichen	Zustand: O: Datei eröffnet. C: Datei geschlossen.
58 - 58	1	Zeichen	KOORID Nach Init: X'00' Nach 1. Open: id, X'40' = NOID

**3.1.4.1.1 Beschreibung eines Protokollsatzes**

Bytes	Länge	Format	Inhalt
0 - 3	4	Hex.	Satzlänge <b>feld</b> ( <i>ℓℓbb</i> )
4 - 7	4	Hex.	Länge Kopf <b>teil</b> ( <i>ℓℓbb</i> )
8 - 13	6	Zeichen	Datum Poolinit (1. Teil Key)
14 - 19	6	Numerisch	Zeit Poolinit (2. Teil Key)
20 - 23	4	Hex.	Fortlaufende Numerierung (3. Teil Key)
24 - 24	1	Zeichen	Satztyp: P
25 - 25	1	Zeichen	Zusatztyp: P = Protokollsatz D = Diagnosesatz
26 - 28	3		Reserve
29 - 29	1	Zeichen	Betriebsart: D: DBH N: TIAM S: UTM - Synchronisiert U: UTM - Unsynchronisiert
30 - 33	4	Numerisch	TSN des Aufrufers
34 - 41	8	Alpha	Transaktion - Id. Betrieb = N: TSN + 4 Spaces S: UTM - interne Kennung U: KCLOGTER / KCBENID
42 - 49	8	Alpha	Applikation oder Spaces
50 - 57	8	Alpha	Hostname oder * + 7 Spaces
58 - 65	8	Alpha	User - id.
66 - 71	6	Numerisch	Uhrzeit der Protokollierung (HHMMSS).
72 - 75	4	Binär	Länge des Datenteils. Format: <i>ℓℓ</i> + 2 Spaces.
76 -			Datenteil: Verbales Kommando Formatiertes Kommando Diagnosetext

### 3.1.5 Beschreibung der Statusdatei

Die Statusdatei wird bei einem Rollback - All für Transaktionen der Betriebsart S oder für Transaktionen, die mit TR,AS begonnen wurden (ab V11), erstellt.

Dies ist eine logisch eigene Datei. Physikalisch ist sie in der BIM - Datei untergebracht.

Format des Satzes: (1 PAM - Page)

Feld	Bytes	Länge	Format	Inhalt
1	0 - 31	32		Vorspann (BIM - Format):
	0 - 3	4	X	- FILE - ID.
	4 - 7	4	X	- Reserve
	8 - 11	4	X	- Version
	12 - 15	4	X	- Pointer - Rück
	16 - 19	4	X	- Pointer - Vor
	20 - 20	1	C	- Kennung (= C'S')
	21 - 27	7	X	- Reserve
	28 - 29	2	B	- Distanz (=0)
30 - 32	2	B	- Anzahl Bytes: 0 = leerer Satz sonst Länge der Daten	
2	32 - 51	20	C	Kennung:
	32 - 32	1	C	- Modus
	33 - 33	1	C	- Origin (= C'U')
	34 - 34	1	C	- Reserve - 1
	35 - 35	1	C	- Reserve - 2
	36 - 43	8	C	- Hostname
	44 - 51	8	C	- Appliname
3	52 -		X	Bei Modus = S:
	52 - 52	1	X	- Session - counter
	53 - 54	2	X	- Vorgang - Nr. 1
	55 - 56	2	X	- Vorgang - Nr. 2
	57 - ...	2	X	... - Vorgang - n ...
4	52 -		C	Bei Modus nicht = S
	52 - 59	8	C	- TID - 1
	60 - 67	8	C	- TID - 2
	68 - ...	8	C	... - TID - n ...

### 3.2 Datei in CISUTM

CISUTM benützt eine PAM - Datei um die Transaktionsinformationen zu speichern. Ein Teil dieser Informationen kann im Memory - Pool stehen. Alles was darüber hinausgeht, kommt in die PAM - Datei. Im Standardfall stehen z.B. die 50 ersten Werte im Speicher und die restlichen 150 in der PAM - Datei. (siehe auch Abschnitt über Memory - Pool in CISUTM auf Seite 28).

Name der Datei: CISUTM.RIOFILE.aaaaaaa wobei aaaaaaa der Anwendungsname ist.

Größe: Pro Transaktion 8 PAM - Pages, also im Standardfall:  $(200 - 50) * 8 = 1.200$  Pages. Die 8 Pages (16 KB) sind wie der Arbeitsbereich in CISUTM aufgebaut.

Falls die Datei noch nicht vorhanden ist, wird sie vom ersten CISUTM mit dem FILE - Kommando eingerichtet. Nach Beenden des letzten Tasks wird sie wieder gelöscht (jedoch nicht wenn sie schon vor dem Starten des 1. Tasks vorhanden war).

Die Größe des Bereichs pro Transaktion steht im Feld KLUSER in CISUTM + X'3558' als Wort (z.B. X'00004000' bei 8 PAM - Pages = 16K).

## 4 Tabellen

### 4.1 Tabellen in CISKOOR

#### 4.1.1 KOOR - Task - Tabelle

Diese Tabelle befindet sich im Common - Memory - Pool auf der 2. Seite (ab Adresse X'1000'). Sie hat Einträge von 8 Bytes. Die Anzahl Einträge ist gleich dem Wert aus dem Parameter KMAXT=.

Die ersten 4 Bytes enthalten die TSN einer KOOR - Task, oder Spaces wenn es den Task nicht gibt. Der erste Task von CISDBH belegt immer den ersten Eintrag.

Die nächsten 4 Bytes werden bei Aufrufen von einem KOOR zu einem anderen für Rückmeldungen benützt.

#### Verwaltung der KOOR - Task - Tabelle:

1. Beim Einrichten des Pools

wird die Tabelle (im ersten Task) angelegt und mit Spaces gelöscht. Der erste Task belegt den ersten Eintrag.

2. Wenn ein Task dazukommt (Automatisches Starten per KNTASKS oder manuelles Starten).

sucht er den ersten freien Eintrag, und belegt diesen. (Diese Routine ist serialisiert).

3. Wenn ein Task sich beendet

löscht er seinen Eintrag mit Spaces. Das automatische Beenden von Tasks mit dem KNTASK - Parameter sucht rückwärts in der KOOR - TSN Tabelle die zu beendenden Tasks aus. Diesen Tasks wird dann eine Nachricht geschickt, daß sie sich beenden sollen.

#### 4.1.2 Partnertabelle in CISKOOR

1. Allgemeines:

Diese Tabelle befindet sich im Common - Memory - Pool gleich hinter der KOOR - Task - Tabelle. Für jeden möglichen Partner gibt es einen Eintrag (KNCOMM= max. Anzahl Partner).

Ein Eintrag hat einen festen Teil und einen variablen Teil mit 8 Byte großen Feldern.

Im Falle des "Inlinked" CISKOOR wird nur ein Eintrag benützt.

Die einzelnen Eintäge werden beim Anmelden ./ Abmelden der Kommunikation angelegt / ausgetragen.

Wird ohne Transaktionen jedoch mit Before - Images gearbeitet, so wird die Position in der TSN - Tabelle als Transaktions - Identifikation für die Before - Images benützt.

## Tabellen

### 2. Einrichten der Tabelle:

Die TSN - Tabelle wird in der Routine ENAMP als Teil des Memory - Pools angelegt. Die Größe der Tabelle wird durch den Parameter KNCOMM bestimmt. Die einzelnen Einträge werden mit Spaces gefüllt.

### 3. Ändern der Größe der Tabelle:

Default - Wert für die Größe ist 100. Um dies zu ändern muß der Parameter KNCOMM=n benützt werden.

### 4. Verwaltung der Tabelle:

Folgende Felder dienen der Verwaltung der Tabelle:

Distanz zu Anfang	POOLATA (DS F):	POOL + X'1C'
Distanz zu 1. freien Eintrag	POOLATP (DS F):	POOL + X'20'
Distanz zu Ende	POOLATE (DS F):	POOL + X'24'

Länge eines Eintrags: X'30' = 48 + (KMAXT \* 8)

### 5. Beschreibung des Partnertabelleneintrags in CISKOOR

Bytes		Länge	Format	Inhalt
dezimal	hex.			
0 - 3	0 - 3	4	C	TSN (Spaces wenn Eintrag leer)
4 - 11	4 - B	8	C	Hostname
12 - 19	C - 13	8	C	Appliname
20 - 27	14 - 1B	8	C	User - id.
28 - 31	1C - 1F	4	B	Anfangsadresse Poolsatz dieser TSN.
32 - 35	20 - 23	4	B	Länge Poolsatz (in Vielfachem von 4K) (z.B. 1 heißt 4096 Bytes)
36 - 37	24 - 25	2	B	Version des Partners (9, 10, 11)
38 - 38	26 - 26	1	C	Betriebsart N = Teilnehmer U = UTM - unsynchronisiert S = UTM - synchronisiert
39 - 39	27 - 27	1		Reserve
40 - 45	28 - 2D	6	C	"Enable" - Zeit
46 - 47	2E - 2F	2		Reserve
48 -	30 -	8	B	Einträge von 8 Bytes pro möglichen KOOR - Task mit: - EI - ID für diesen KOOR - Task - REFNUM für diesen KOOR - Task

### Verwaltung der Partnertabelle

Jeder Partner, der sich anmeldet, bekommt einen Eintrag in der Partnertabelle zugewiesen. Wenn sich ein Partner abmeldet, so wird der Eintrag frei.

Die Felder am Ende des Eintrags enthalten die Kurzkenung für die Kommunikation zu diesem Partner (EI - Kennung vom Event - Item in der Richtung CISKOOR zu CISI). Die Position dieser Felder entspricht den Positionen in der KOOR - Task - Tabelle. So benützt z.B. der erste KOOR - Task in jedem Partnereintrag das erste Feld mit der Kurzkenung.

Diese Felder werden folgendermaßen verwendet:

1. Anmelden eines neuen Partners.

Der Partner meldet sich bei einem der KOOR - Tasks an. Dieser Task kennt dann die Kurzkenung und speichert sie gleich im entsprechenden Feld in der Partnertabelle. Alle anderen Felder für Kurzkenungen enthalten noch binäre Nullen. Wenn in Zukunft der Partner wieder mit diesem Task verkehrt, so ist die Kurzkenung zum Rückmelden schon bekannt.

2. Der Partner spricht einen anderen Task an.

Dieser Task stellt beim Senden der Antwort fest, daß er noch keine Kurzkenung in "seinem" Feld gespeichert hat. Er meldet das Event - Item an und speichert die Kurzkenung für zukünftige Aufrufe.

3. Ein Partner meldet sich ab.

Der Partner meldet sich bei einem bestimmten DBH Task ab. Dieser Task kann die Verbindung auflösen. Gibt es noch andere Felder, die eine Kurzkenung enthalten, so sendet der KOOR - Task diese an die entsprechenden anderen KOOR - Tasks, damit sie auch die Verbindung abbauen. Anschließend kann der ganze Eintrag von diesem Partner freigegeben werden.

#### 4.1.3 Beschreibung der TID - Tabelle in CISKOOR / CISBIM

1. Allgemeines:

Die TID - Tabelle enthält pro aktiver Transaktion einen Eintrag. Der Eintrag wird bei Transaktionsanfang angelegt und bei Transaktionsende gelöscht (mit Spaces gefüllt). Jeder Eintrag enthält einen Teil für CISKOOR/CISBIM und einen zweiten Teil für CISBIM. Die Position in der TID - Tabelle ist die Kennung für das Modul CISBIM.

2. Einrichten der Tabelle:

TID - Tabelle wird in der Routine VORBE mit REQM angelegt. Die Größe der Tabelle wird durch den Parameter KNTRANS bestimmt. Die einzelnen Einträge werden mit Spaces initialisiert.

3. Ändern der Größe der Tabelle:

Default - Wert für die Größe ist 200. Um den Wert zu ändern muß der Parameter KNTRANS = n benützt werden.

4. Verwaltung der Tabelle:

Folgende Felder dienen der Verwaltung der Tabelle:

Anfangsadresse	POOLAIA	(DS A) : POOL
Endadresse + 1	POOLAIE	(DS A) : POOL
1. freier Eintrag	POOLAIP	(DS A) : POOL

Länge eines Eintrags      X'60' = 96

## 5. Beschreibung des TID - Tabellen - Eintrags in CISKOOR

Bytes		Länge	Format	Inhalt
dezimal	hex.			
0 - 1	0 - 1	2	B	ID (X'0001' - X'FFFF')
Felder für CISKOOR und CISBIM				
2 - 9	2 - 9	8	C	Hostname (Transaktion)
10 - 17	A - 11	8	C	Appliname (Transaktion)
18 - 25	12 - 19	8	C	TID (Transaktion)
26 - 33	1A - 21	8	C	Hostname (Partner)
34 - 41	22 - 29	8	C	Appliname (Partner)
42 - 49	2A - 31	8	X	UTM - S - TID
50 - 53	32 - 35	4	C	TSN
54 - 54	36 - 36	1	C	Betriebsart N = Teilnehmer U = UTM - unsynchronisiert S = UTM synchronisiert
55 - 57	37 - 39	3		Reserve
58 - 65	3A - 41	8	C	User
Felder für CISBIM				
66 - 66	42 - 42	1	X	Kennungen X'80' : Externe TA X'40' : Interne TA X'20' : TA in Peta X'10' : TA mit Status X'08' : BIM - Daten gespeichert X'04' : TA in Peta - Warmstart
67 - 67	43 - 43	1		Reserve
68 - 71	44 - 47	4	B	Distanz in PAM - Page
72 - 75	48 - 4B	4	B	1. Pam - Page
76 - 79	4C - 4F	4	B	Aktuelle Pam - Page
80 - 83	50 - 53	4	B	Vorhergehende PAM - Page
84 - 87	54 - 57	4	B	Letzte PAM - Page
88 - 91	58 - 5B	4	B	Anzahl PAM Pages
92 - 95	5C - 5F	4	B	Version



## 5. Beschreibung des TID - Tabellen - Eintrags in CISSP

Bytes		Länge		Format	Inhalt
dezimal	hex.	dez.	hex.		
0 - 23	0 - 17	24	18	C	Kennung: 8 Bytes Hostname 8 Bytes Appliname 8 Bytes TID
24 - 77	18 - 4D	54	36	C	Dateiname der HD (vollqualifiziert)
78 - 78	4E - 4E	1	1	C	Kennung - 0: Space = initialisiert H = ganze HD gesperrt R = Datei reserviert S = Datei secured
79 - 79	4F - 4F	1	1	C	Kennung - 1: A = aufsteigend R = random V = verdichtet
80 - 83	50 - 53	4	4		Reserve
84 - 87	54 - 57	4	4	B	Anzahl gesperrter Sätze
88 - 91	58 - 5B	4	4	B	Adresse 1. Zelle im Pool oder Null
92 - 95	5C - 5F	4	4	B	Adresse letzte Zelle im Pool oder Null
96 - 99	60 - 63	4	4	B	Anzahl belegter Zellen im Pool oder Null
100 - 101	64 - 65	2	2	B	EAM - Dateiname oder Null
102 - 103	66 - 67	2	2	B	Letzte Blocknummer in EAM oder Null
104 - 107	68 - 6B	4	4	B	1. Pam - Page oder Null
108 - 111	6C - 6F	4	4	B	Letzte PAM - Page oder Null
112 - 115	70 - 73	4	4	B	Anzahl PAM - Pages oder Null

## 4.1.5 Beschreibung des Poolkopfes

Bytes		Länge		Format	Inhalt
dezimal	hex.	dez.	hex.		
0 - 3	0 - 3	4	4	B	Distanz zu Anfang TID - Tabelle
4 - 7	4 - 7	4	4	B	Distanz zu Ende TID - Tabelle
8 - 11	8 - C	4	4	B	Distanz zu nächstem freien Eintrag
12 - 15	C - F	4	4	B	Länge TID - Eintrag
16 - 19	10 - 13	4	4	B	Länge Zelle
20 - 23	14 - 17	4	4	B	Distanz zu Anfang Sperrlisten
24 - 27	18 - 1B	4	4	B	Distanz zu Ende Sperrlisten
28 - 31	1C - 1F	4	4	B	Distanz zu nächstem freien Eintrag
32 - 35	20 - 23	4	4	B	Anzahl freie Einträge
36 - 39	24 - 27	4	4	B	Datei: Free list
40 - 43	28 - 2B	4	4	B	Datei: Nächste PAM - Page
44 - 63	2C - 3F	20	14		Reserve

#### 4.1.6 Beschreibung des Pools der Sperrlisten in CISSP

##### 1. Allgemeines:

Der Pool der Sperrlisten wird von CISKLOOR im Arbeitsspeicher eingerichtet.

Der Pool ist in Zellen fester Größe unterteilt. Die einzelnen Zellen sind durch ihr jeweiliges erstes Wort vorwärts verkettet. Somit besteht jede Zelle aus einem Wortpointer und x Worten Text. Ein Pointer mit Inhalt Null ist das Ende der Kette.

Es gibt für jede Sperrliste eine Kette von Zellen. Die Anfangsadresse der Kette (= Adresse der ersten Zelle) steht im TID - Tabelleneintrag. Zusätzlich gibt es noch die Kette der freien Zellen.

##### 2. Einrichten des Pools:

Der Pool wird von CISKLOOR angelegt. Am Anfang steht ein Poolkopf mit Verwaltungsdaten. Am Ende steht die TID - Tabelle. Im Multitask - Betrieb ist der Pool von allen Tasks adressierbar. Da nur ein Vielfaches von 4K belegt werden kann, wird der überschüssige Platz nach folgendem Schema dem Pool zugewiesen:



##### 3. Ändern der Größe des Pools:

Soll die Größe des Pools geändert werden, so muß der CISKLOOR - Parameter KLOCKP angegeben werden. Er steht standardmäßig auf 100 KB.

##### 4. Verwaltung des Pools:

Folgende Felder dienen der Verwaltung des Pools:

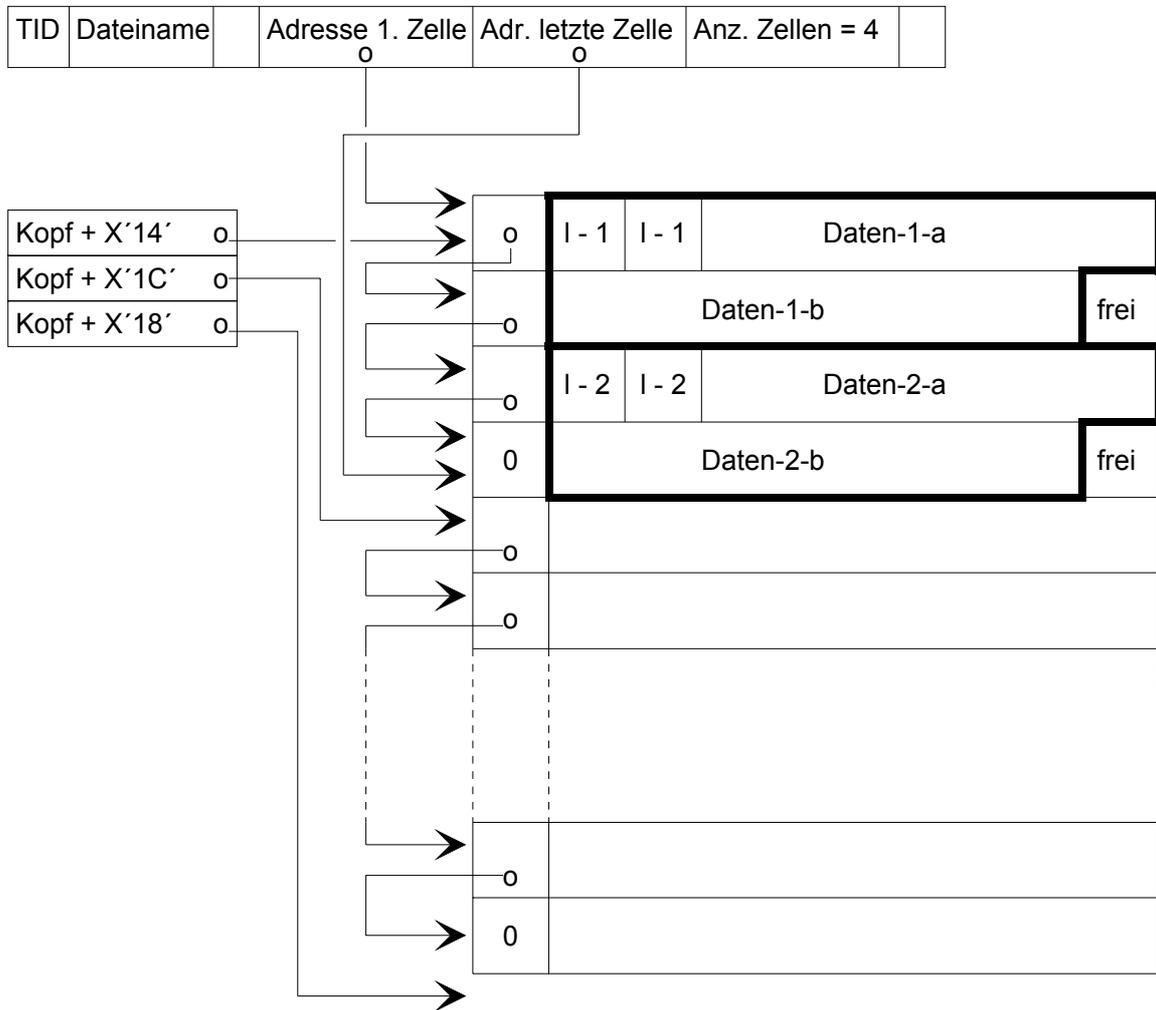
Adresse Pool:    VCISSP.FALOCK:    L'00024'  
 Länge Pool:     VCISSP.FLLOCK:     L'00028'  
 Anzahl Einträge: VCISSP.FNLOCK:    L'0002C'

(Siehe auch Parameter KLOCKP)



# Tabellen

## 7. Beispiel einer Pool - Belegung



## 4.2 Tabellen in CISDBH

### 4.2.1 DBH - Task - Tabelle

Diese Tabelle befindet sich im Common - Memory - Pool auf der 2. Seite (ab Adresse X'1000'). Sie hat Einträge von 8 Bytes. Die Anzahl Einträge ist gleich dem Wert aus dem Parameter DMAXT=.

Die ersten 4 Bytes enthalten die TSN einer DBH - Task, oder Spaces wenn es den Task nicht gibt. Der erste Task von CISDBH belegt immer den ersten Eintrag.

Die nächsten 4 Bytes werden bei Aufrufen von einem DBH zu einem anderen für Rückmeldungen benützt.

#### Verwaltung der DBH - Task - Tabelle:

1. Beim Einrichten des Pools

(im ersten Task) wird sie angelegt und mit Spaces gelöscht. Der erste Task belegt den ersten Eintrag.

2. Wenn ein Task dazukommt

(automatisches Starten per DNTASKS oder manuelles Starten) sucht er den ersten freien Eintrag, und belegt diesen. (Diese Routine ist serialisiert).

3. Wenn ein Task sich beendet

löscht er seinen Eintrag mit Spaces. Das automatische Beenden von Tasks mit dem DNTASK - Parameter sucht rückwärts in der DBH - TSN - Tabelle die zu beendenden Tasks aus. Diesen Tasks wird dann eine Nachricht geschickt, daß sie sich beenden sollen.

### 4.2.2 Partnertabelle in CISDBH

Diese Tabelle befindet sich im Common - Memory - Pool gleich hinter der DBH - Task - Tabelle. Für jeden möglichen Partner gibt es einen Eintrag (DNCOMM= max. Anzahl Partner).

Ein Eintrag hat einen festen und einen variablen Teil mit 8 Byte großen Feldern.

## Tabellen

Beschreibung eines Eintrags:

Bytes		Länge	Format	Inhalt
dezimal	hex.			
0 - 3	0 - 3	4	C	TSN (Spaces wenn Eintrag leer)
4 - 11	4 - B	8	C	User - id.
12 - 15	C - F	4	b	Anfangsadresse Poolsatz dieser TSN
16 - 19	10 - 13	4	B	Länge Poolsatz (in Vielfachem von 4K)
20 - 20	14 - 14	1	C	Betriebsart: N = Teilnehmer U = UTM - unsynchronisiert S = UTM - synchronisiert
21 - 21	15 - 15	1	C	Kommunikation: D = DCAM P = P1 - Eventing
22 - 22	16 - 16	1	B	Session - counter (von UTM)
23 - 23	17 - 17	1		Reserve
24 - 31	18 - 1F	8	C	Hostname
32 - 39	20 - 27	8	C	Applikationsname
40 - 45	28 - 2D	6	C	"Enable" - Zeit
46 - 47	2E - 2F	2	B	Version des Partners (9, 10, 11)
48 -	30 -	8	B	Einträge von 8 Bytes pro möglichen DBH - Task: - EI - ID für diesen DBH - Task - REFNUM für diesen DBH - Task

### Verwaltung der Partnertabelle

Jeder Partner, der sich anmeldet, bekommt einen Eintrag in der Partnertabelle zugewiesen. Wenn sich ein Partner abmeldet, so wird der Eintrag frei.

Die Felder am Ende des Eintrags enthalten die Kurzkenung für die Kommunikation zu diesem Partner (EI - Kennung vom Event - Item in der Richtung CISDBH zu CISCON). Die Position dieser Felder entsprechen den Positionen in der DBH - Task - Tabelle. So benützt z.B. der erste DBH - Task in jedem Partnereintrag das erste Feld mit der Kurzkenung.

Diese Felder werden folgendermaßen verwendet:

1. Anmelden eines neuen Partners:

Der Partner meldet sich bei einem der DBH - Tasks an. Dieser Task kennt dann die Kurzkenung und speichert sie im entsprechenden Feld in der Partnertabelle. Alle anderen Felder für Kurzkenungen enthalten noch binäre Nullen. Wenn in Zukunft der Partner wieder mit diesem Task verkehrt, so ist die Kurzkenung zum Rückmelden schon bekannt.

2. Der Partner spricht einen anderen Task an:

Dieser Task stellt beim Senden der Antwort fest, daß er noch keine Kurzkenung in "seinem" Feld gespeichert hat. Er meldet das Event - Item an und speichert die Kurzkenung für zukünftige Aufrufe.

3. Ein Partner meldet sich ab:

Der Partner meldet sich bei einem bestimmten DBH Task ab. Dieser Task kann die Verbindung auflösen. Gibt es noch andere Felder, die eine Kurzkenung enthalten, so sendet der DBH - Task diese an die entsprechenden anderen DBH - Tasks, damit sie auch die Verbindung abbauen. Anschließend kann der ganze Eintrag von diesem Partner freigegeben werden.

### 4.3 Tabellen in CISCON

Die Kommunikation zwischen CISCON und CISDBH wird in CISCON durch folgende Tabellen gesteuert:

- Transaktionstabelle
- Paßworttabelle
- DBH - Tabelle

Bei Betriebsart = S wird keine Transaktionstabelle benützt, sondern der DB - spezifische Bereich im TAM.

Zum Entschlüsseln der CIS - Kommandos enthält CISCON eine

- Kommandotabelle

#### 4.3.1 Transaktionstabelle

Für jede aktive Transaktion gibt es einen Eintrag in dieser Tabelle mit folgendem Inhalt:

Feld	Bytes		Länge	Inhalt
	dezimal	hex.		
DTTTI	0 - 7	0 - 7	8	Transaktionsidentifikation: - Teilhaber 8 - stelliger Terminalname - Teilnehmer TSN plus 4 Bytes
DTTPW	8 - 13	8 - D	6	Aktuelles CIS - Paßwort
DTTK	14 - 14	E - E	1	Kennungen: DTTKTA: X'01' - eine Transaktion ist angefangen
	15 - 15	F - F	1	
DTTDIT	16 - 17	10 - 11	2	Distanz in Tabelle der DBH's für DBH in dem TR,A bearbeitet wurde (nur gültig wenn DTTK = DTTKTA)
DTTDI	18 - 19	12 - 19	2	Distanz in Tabelle der DBH's für aktuellen DBH.

Zusätzliche Namen/Felder: DTT Name der DSECT  
ELDTT Länge eines Eintrags (20 = X'14')

ATABTA Anfangsadresse der Tabelle: CISCON + X'4978'  
 ATABTE Endadresse + 1 der Tabelle: CISCON + X'497C'  
 ATABTF Adresse des 1. freien Eintrags : CISCON + X'4980'  
 ANZTABT Anzahl Einträge (DS F): CISCON + X'39FC'

Bemerkungen: Bei Mode = Single wird diese Tabelle nicht benützt.  
 Bei Betrieb = S wird nicht diese Tabelle, sondern der DB - spezifische Bereich im TAM benützt (siehe nächste Seite).

#### 4.3.2 DB - spezifischer Bereich im TAM

Distanz im TIAM	Länge	Name in CISCO 1	Inhalt
X'20'	8	TAMARTK	Terminalkennung: KCBENID wenn ≠ Spaces sonst KCLOGTER
X'28'	1	TAMARK	Kennung zur Terminalkennung: L = in TAMARTK steht die KCLOGTER B = in TAMARTK steht die KCBENID
X'29'	1	TAMARSN	Session - Nummer aus TAMSESCN
X'2A'	2		Reserve
X'2C'	4	TAMARTSN	TSN
X'30'	6	TAMPW	Aktuelles Paßwort
X'36'	6	TAMPWT	Paßwort bei TR,A
X'3C'	4		Reserve
X'40'			

#### 4.3.3 Paßworttabelle

Für jedes mögliche Paßwort in der Datei gibt es einen Eintrag in dieser Tabelle mit folgendem Inhalt:

Feld	Bytes		Länge	Inhalt
	dezimal	hex.		
DTPPW	0 - 5	0 - 5	6	Paßwort * im ersten Eintrag bedeutet: "Alle Paßwörter" (Mode = Single)
DTPZR	6 - 13	6 - D	8	Zielrechner: Name des fremden Rechners * bedeutet: eigener Rechner.
DTPZA	14 - 14	E - E	1	DBH - Kennung
	15 - 15	F - F	1	Reserve
DTPI	16 - 17	10 - 11	2	Distanz in der DBH - Tabelle

Zusätzliche Namen/Felder: DTP Name der DSECT  
ELDTP Länge eines Eintrags (18 = X'12')

ATABPA Anfangsadresse der Tabelle: CISCON + X'496C'  
 ATABPE Endadresse +1 der Tabelle: CISCON + X'4970'  
 ANZTABP Anzzahl Einträge in der Tab.(DS F): CISCON + X'4974'

Bemerkungen: Die Paßworte stammen aus: CISCON - Parameter DIST= oder  
 CISDBH - Parameter DPASS=

#### 4.3.4 DBH - Tabelle

Für jeden DBH, mit dem kommuniziert wird, existiert ein Eintrag in dieser Tabelle mit folgendem Inhalt:

Feld	Bytes		Länge		Inhalt
	dezimal	hex.	dez.	hex.	
DTDZR	0 - 7	0 - 7	8	8	Zielrechner: Name des fremden Rechners * bedeutet: eigener Rechner.
DTSZA	8 - 8	8 - 8	1	1	DBH - Kennung
DTDMO	9 - 9	9 - 9	1	1	Modus: S = Single : DBH= A = Automatic : PARTNER= D = Distribution : DIST=
	10 - 11	A - B	2	2	Reserve
DTDZC	12 - 12	C - C	1	1	Zustand der Verbindung: I : Initial C : Connection L : Lost
	13 - 15	D - F	3	3	Reserve
DTDAD	16 - 19	10 - 13	4	4	Adresse für das Protokoll zwischen CISCON und CISDBH.
DTDSAV	20 - 67	14 - 43	48	30	Zum Speichern Kennungen und Infos über Kommunikation (siehe nächste Seiten).

Zusätzliche Namen/Felder: DTD Name der DSECT  
ELDTD Länge eines Eintrags (68 = X'44')

ATABDA Anfangsadresse der Tabelle : CISCON + X'4960'  
 ATABDE Endadresse + 1 der Tabelle : CISCON + X'4964'  
 ANZTABD Anzahl Einträge in der Tabelle (DS F) : CISCON + X'4968'

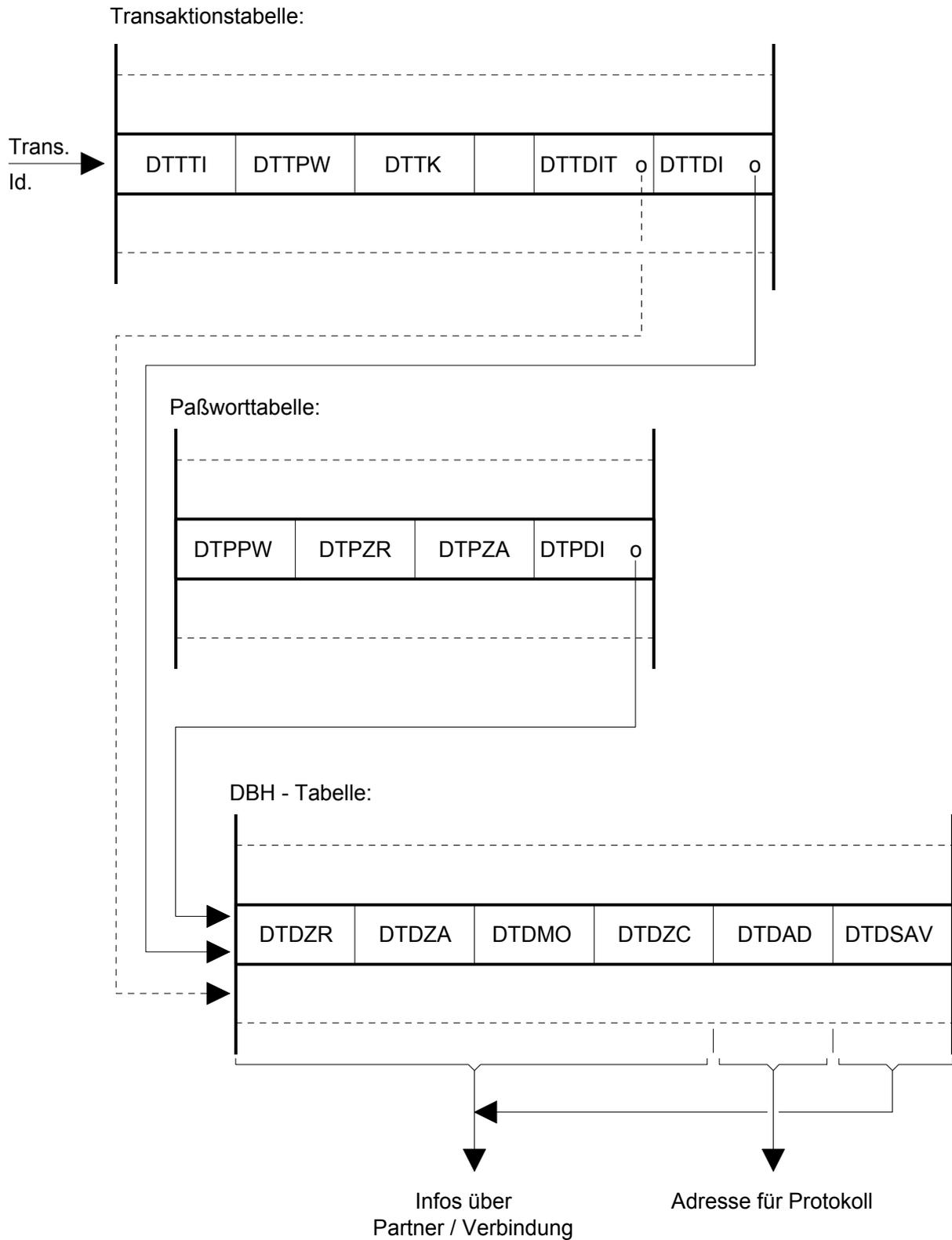
## 4.3.4.1 Inhalt von DTDSAV bei Kommunikation über DCAM

Bytes		Länge		Feldname im Modul DCAM	Inhalt
dezimal	hex.	dez.	hex.		
20 - 23	14 - 17	4	4	DCAMP	Adresse Poolsatz wird mit REQM 9 in der Routine OPEN angelegt. 1 Bereich für alle DBH's
24 - 27	18 - 1B	4	4	FCID	Aktuelle CID mit YSHOWCB aus CCB nach YOPNCON gewonnen.
28 - 29	1C - 1D	2	2	FMAXLN	Maximale Nachrichtenlänge
30 - 31	1E - 1F	2	2		Reserve
32 - 32	20 - 20	1	1	DCKA	Zustand der Anwendung: I : Initial O : Open E : Error
33 - 33	21 - 21	1	1	DCKC	Zustand der Verbindung: I : Initial O : Open D : Disconnected E : Error L : Lost (Try later)
34 - 34	22 - 22	1	1	DCKS	Etwas gesendet? Y : Yes N : No
35 - 35	23 - 23	1	1		Reserve
36 - 67	24 - 43	32	20		Nicht benützt

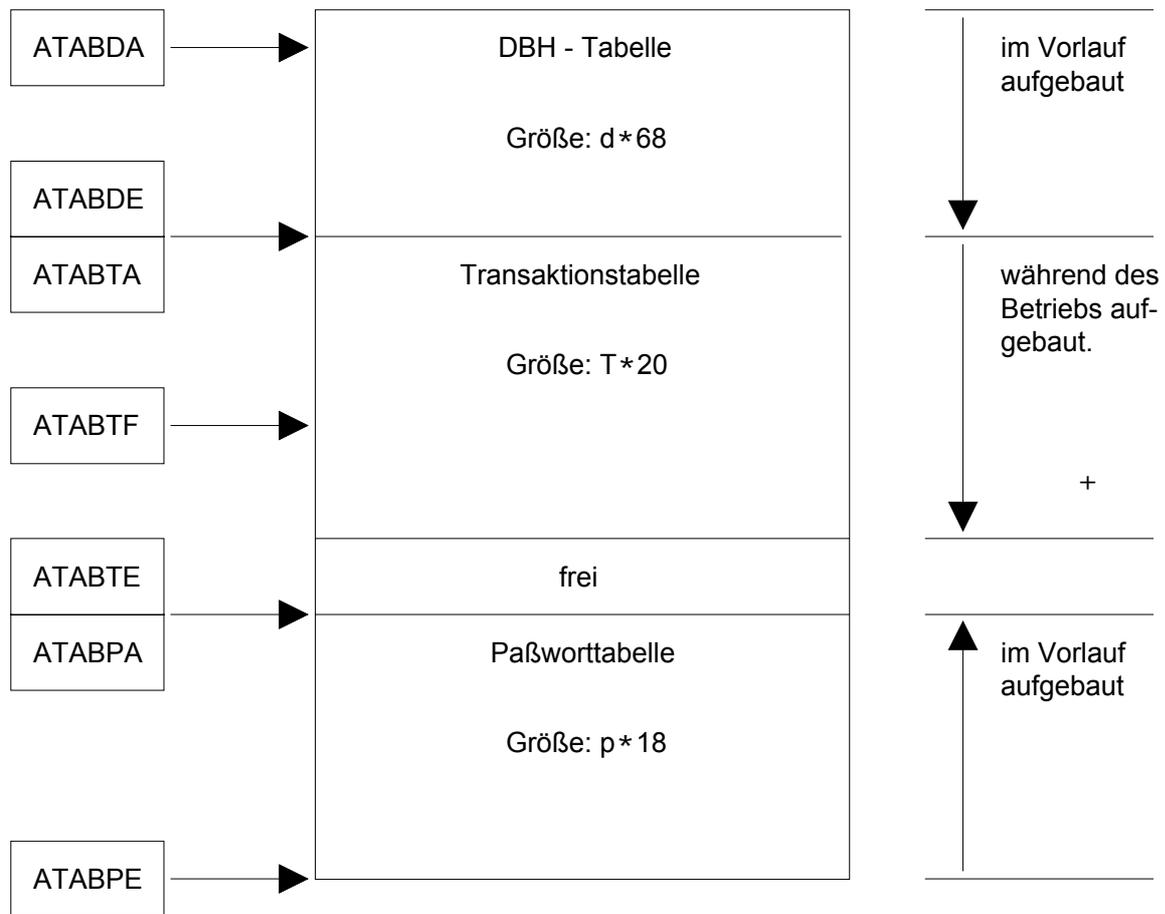
**4.3.4.2 Inhalt von DTDSAV bei Kommunikation über P1 - Eventing**

Bytes		Länge		Feldname im Modul DCAM	Inhalt
dezimal	hex.	dez.	hex.		
20 - 23	14 - 17	4	4	CODBHN	DBH - Name, Format DBHI I = DBHID
24 - 27	18 - 1B	4	4	COEIID1	Kurzkenung Event - Item 1 CISCON —▶ CISDE
28 - 31	1C - 1F	4	4	COREFN1	Referenznummer Event - Item 1 CISCON —▶ CISDE
32 - 35	20 - 23	4	4	COEIID2	Kurzkenung Event - Item 2 CISDBH —▶ CISC
36 - 39	24 - 27	4	4	COREFN2	Referenznummer Event - Item 2 CISDBH —▶ CISC
40 - 43	28 - 2B	4	4	COMPID	Kurzkenung Memory - Pool
44 - 47	2C - 2F	4	4	COAMP1	Anfangsadresse des Memory - Pools
48 - 51	30 - 33	4	4	COAMP2	Adressbereich im Memory - Pool
52 - 55	34 - 37	4	4	COREQS	MP - Size
56 - 56	38 - 38	1	1	COCOM	Kommunikation aufgebaut ? (J/N)
57 - 57	39 - 39	1	1	COCOM2	1. Kommunikation ? (J/N)
58 - 58	3A - 3A	1	1	COSEND	Etwas vorher gesendet ? (J/N)
59 - 59	3B - 3B	1	1	COWAIT	Warten auf Antwort ? (J/N)
60 - 60	3C - 3C	1	1	COWART	Warten gleich nach Senden ? (J/N)
61 - 63	3D - 3F	3	3		Reserve
64 - 65	40 - 41	2	2	COID	ID (in der TSN - Tabelle in CISDBH)
66 - 67	42 - 43	2	2		Nicht benützt.

### 4.3.5 Verkettung der Tabellen



#### 4.3.6 Organisation der Tabellen im Speicher



Erläuterungen: d = Anzahl Einträge in DBH - Tabelle      ANZTABD (DS F)  
 t = Anzahl Einträge in Transaktionstabelle      ANZTABT (DS F)  
 p = Anzahl Einträge in Paßworttabelle      ANZTABP (DS F)

Das Wort KLBER (im Modul CISCO1) enthält die Größe des gesamten Bereichs (2 \* 4K).

## 4.3.7 Kommandotabelle in CISCON (Tabelle TKOM)

Kommando + Ergänzung	Typ	Anzahl Param	Spezialkennun g
AE A	2	5	0
AE F	0	4	0
AE K	2	5	0
AE KD	2	5	0
AE KF	0	4	0
AE KS	2	5	0
AK	1	5	0
BI	1	5	0
BL	0	4	0
CL A	0	4	7
CL	0	4	0
D[R]	0	4	0
EI	4	5	0
EN	0	4	0
EX	0	4	0
F[R]	0	4	0
G[E] FAA	3	5	0
G[E] FBA	3	5	0
G[E] FEA	3	5	0
G[E] KF	3	5	0
G[E] KFS	3	5	0
G[E] KFN	3	5	0
G[E] KFNS	3	5	0
G[E] KFR	3	5	0
G[E] KFRS	3	5	0
G[E] sonst.	5	6	0
HI	3	5	0
I[G] *	0	4	0
I[G] F	1	5	0
I[G] I	0	4	0
I[G] K	0	4	0
I[G] S	1	5	0
I[G] SP	0	4	0
I[G] Z	0	4	0
K[O]	4	5	0
L[O]	0	4	0
N[I]	1	5	0
OD	1	5	0
PU A	2	5	0
PU F	0	4	0
PU K	2	5	0
PU KF	0	4	0
R[E]	0	4	0
SE	0	4	0
SI	0	4	0
SO	1	5	0
SP	0	4	0

Tabellen

Kommando + Ergänzung	Typ	Anzahl Param	Spezialkennun g
ST	1	5	0
SU	1	5	0
SY	3	5	0
TE	0	4	0
TR A	0	4	5
TR AL	0	4	5
TR AS	0	4	5
TR E	0	4	6
TR EB	0	4	6
TR PE	0	4	4
TR R	0	4	6
TR RB	0	4	6
TR SA	0	4	4
TR SL	0	4	4
TR UA	0	4	4
TR UC	0	4	4
TR UE	0	4	4
TR UL	0	4	4
TR UP	0	4	4
TR UR	0	4	4
TR WE	0	4	4
TR WR	0	4	4
U[N]	1	5	0
V[E]	1	5	9
WA	3	5	0
WI	3	5	0
WR	0	4	0
ZE[E] M	4	5	0
ZE[E] sonst.	3	5	0
\$A	0	4	A
\$D	3	5	8
\$P A	0	4	1
\$P D	0	4	4
\$P E	0	4	2
\$P R	0	4	4
\$T A	0	4	3
\$T D	0	4	4
\$T E	0	4	4

## Kommandotypen

Typ	Bedeutung
0	4 Parameter
1	5 Parameter: 5. Parameter = AZI / ASK
2	5 Parameter 5. Parameter = A / KAP (Hin)
3	5 Parameter 5. Parameter = Bereich (Zurück)
4	5 Parameter 5. Parameter = Bereich (Hin) Bereich (Zurück)
5	6 Parameter 5. Parameter = LZI 6. Parameter = ZI

## Spezialkennungen:

Spezialkennung	Bedeutung
0	Normal Kommando
1	\$P,A
2	\$P,E
3	\$T,A
4	\$T,E / \$T,D
5	TR,A
6	TR,E / TR,R
7	CL,A
8	\$D
9	VERBINDE
A	\$A (ignorieren)

## 4.4 Tabellen in CISUTM

### 4.4.1 Tabelle der Transaktionen

Bytes		Länge		Inhalt
dezimal	hex.	dez.	hex.	
0 - 23	0 - 17	24	18	Transaktionskennung 8 Bytes Hostname 8 Bytes Applikationsname 8 Bytes TID 8 Bytes
24 - 27	18 - 1B	4	4	TSN - Ursprung
28 - 31	1C - 1F	4	4	Distanz im RIO - Bereich oder Null (wenn in Datei).
32 - 33	20 - 21	2	2	PAM - Page (wenn in Datei, sonst Null)
34 - 34	22 - 22	1	1	Flags: x'80' set Bereich im Speicher Distanz angeben reset Bereich in Datei PAM - Page angeben x'40' set 1. Mal reset weitere Male
35 - 35	23 - 23	1	1	Betriebsart:
36 - 43	24 - 2B	8	8	TID
44 - 51	2C - 33	8	8	UTM - User
52 - 55	34 - 37	4	4	Bisherige TSN (von CISUTM)
56 - 59	38 - 3B	4	4	Aktuelle TSN (von CISUTM)

## 5 Spezielle Kommandos

### 5.1 Paßwortschutz

Der Anwender kann ein Paßwort definieren, um an einem bestimmten Terminal \$D - Kommandos zuzulassen.

1. Definition des Paßwortes: Das Paßwort wird in CISVARI + X'18F' definiert. Es ist 8 Bytes lang. Acht Spaces bedeuten "kein Paßwort".

2. Anmelden des Terminals:

\$D P=C´ . . . ´ [ ,DB .XXXXXX ]	abdruckbar
\$D P=X´ . . . ´ [ ,DB .XXXXXX ]	sedezimal

Es können 1 bis 8 Bytes angegeben werden. Sind es weniger als 8 Bytes, so wird mit Space (x'40') ergänzt.

Im Teilhaberbetrieb muß sich jedes Terminal anmelden.

3. Abmelden des Terminals:

\$D P=NONE [ ,DB .XXXXXX ]
----------------------------

4. Bemerkung: CIS wird ohne Paßwort ausgeliefert. Alle Terminals können dann alle \$D - Kommandos eingeben.

## 5.2 Kommandos für CIS

### 5.2.1 Übersicht

\$D CALLTRONL[ ,DB.XXXXXX ]	Einschalten Protokollierung auf SYSLST.
\$D CALLTRONO[ ,DB.XXXXXX ]	Einschalten Protokollierung auf SYSOUT.
\$D CALLTROFF[ ,DB.XXXXXX ]	Ausschalten Protokollierung.
\$D CMEM[ ,DB.XXXXXX ]	Speicherbelegung
\$D CTIMEON[ ,DB.XXXXXX ]	Einschalten Zeitmessung neu.
\$D CTIMEOLD[ ,DB.XXXXXX ]	Einschalten Zeitmessung alt.
\$D CTIMEOFF[ ,DB.XXXXXX ]	Ausschalten Zeitmessung.
\$D CTRON[ ,DB.XXXXXX ]	Einschalten SVC - Trace.
\$D CTROFF[ ,DB.XXXXXX ]	Ausschalten SVC - Trace.
\$D CWRITE[ ,DB.XXXXXX ]	Read - only Module überschreibbar machen.

**5.2.2 CALLTRx Protokollierung der CIS Aufrufe**

```

$D CALLTRONL[ ,DB .XXXXXX]
$D CALLTRONO[ ,DB .XXXXXX]
$D CALLTROFF[ ,DB .XXXXXX]

```

```

Einschalten auf SYSLST
Einschalten auf SYSOUT
Ausschalten

```

Bei jedem Aufruf von IVMD wird folgendes protokolliert:

Anzahl Parameter

Inhalt von KL

Inhalt von K (Steht ein \* hinter K, so wurde der Text in Großbuchstaben übersetzt.)

Beispiel einer Protokollierung:

```

KL: 18 ANZAHL PARAMETER: 5
K : SU EDV#0 ,DB .FMULTI

```

```

KL: 14 ANZAHL PARAMETER: 5
K : Z , 2 , T , EDV NAME

```

```

KL: 6 ANZAHL PARAMETER: 4
K : BL , $ , R

```

```

KL: 14 ANZAHL PARAMETER: 5
K : Z , $ , T , KEY NAME

```

```

KL: 12 ANZAHL PARAMETER: 5
K : $D CALLTROFF

```

Diese Protokollierung ist als Test- und Diagnosewerkzeug gedacht. Es wird in die eventuell schon vorhandene SYSLST - Datei geschrieben. (Gemischt mit den Ergebnissen des Drucke - Kommandos).

### 5.2.3 CMEM Speicherbelegung

```
$D CMEM[ ,DB .XXXXXX ]
```

Antwort am Terminal:

Die ersten 4 Zeilen bringen allgemeine Werte in KB. Ab der 5. Zeile folgt eine Liste von belegten und unbelegten Seiten (4KB - Seiten).

U: . . . . . 0 / . . . . . 242 bedeutet:

```
U      : "Used" (=belegt)
      0 : 1. belegte Seite (dezimal)
     180 : Anzahl belegter Seiten (dezimal)
          = 720 KB
```

F: . . . . . 180 / . . . . . 14 bedeutet:

```
F      : "Free" (=frei)
     180 : 1. freie Seite (dezimal)
     14  : Anzahl freier Seiten (dezimal)
          = 56 KB
```

U: . . . . . 194 / xxxxxxxx usw.

### 5.2.4 CTIMEX Zeitmessung

```
$D CTIMEON[ ,DB .XXXXXX ]
$D CTIMEOLD[ ,DB .XXXXXX ]
$D CTIMEOFF[ ,DB .XXXXXX ]
```

Einsch. Zeitmessung neu  
Einsch. Zeitmessung alt  
Ausschalten

Vor dem Verlassen von CISI wird für jedes Kommando ein Satz protokolliert. Seit CIS V11 wird dieser Satz in die Datei mit Namen "CIS.TIME.tttt" (tttt = TSN des Jobs) geschrieben. Diese Datei muß mit dem Programm CISTA ausgewertet werden.

Wird mit CTIMEOLD eingeschaltet, so wird wie in CIS V10 der Satz in die SYSLST01 - Datei geschrieben.

**5.2.5 CTRx      SVC - Trace in CIS**

```
$D CTRON[ ,DB .XXXXXX ]
$D CTROFF[ ,DB .XXXXXX ]
```

Einschalten  
Ausschalten mit Statistik

Die meisten SVCs werden protokolliert, wenn sie ausgeführt werden sollen. Hierzu ist mit dem \$D CTRON - Kommando eine spezielle STXIT - Routine einzuschalten.

Es wird folgendes protokolliert:

Kennung	Kennung der DASI - Sätze (nur wenn mit DASI gearbeitet wird, sonst Spaces).
SVC - NAME	Assembler - Name des SVC's.
ADR.	Adresse im Programm.
LINKNAM	LINK - Name bei Dateizugriffen.
HPAGE	Nummer der Pam - Page bei EAM.
DATEINAME/...	Dateiname bei Datenzugriffen. Der ISAM - Key wird in der / den folgenden Zeilen ausgegeben. Ist er abdruckbar wird er in seiner echten Länge ausgedruckt (max.2 Zeilen). Ist er es nicht, so wird er sedezimal aufbereitet in seiner echten Länge ausgedruckt.(max. 4 Zeilen).

Am Ende wird eine Übersicht der aufgerufenen SVCs gedruckt.

Diese Protokollierung ist als Test- und Diagnosewerkzeug gedacht. Es wird in die SYSLST02 - Datei geschrieben.

**5.2.6 CWRITE      Read - only Module überschreibbar machen**

```
$D CWRITE
```

In CIS werden alle Seiten im Klasse - 6 - Speicher mit dem CSTAT - Makro überschreibbar gemacht.

Dies kann nötig werden, wenn IDA - Kommandos AT oder MOVE gegeben werden sollen.

Es gibt keine Möglichkeit, die Seiten wieder "read - only" zu machen.

## 5.3 Kommandos für CISCON

### 5.3.1 Übersicht

\$D CCDIA x	Diagnose einschalten
\$D CCLST D[,p]	Auflistung der DBH - Tabelle
\$D CCLST P[,p]	Auflistung der Paßworttabelle
\$D CCLST T[,p]	Auflistung der Transaktionstabelle
\$D CCMEM	Speicherbelegung
\$D CCSYS /Kommando	BS2000 - Kommando
\$D CCT=tttt,...	Angabe einer TSN

### 5.3.2 CCDIA      Diagnose schalten

```
$D CCDIA x
```

Schalten der Diagnose. x kann folgende Werte annehmen:

J        : Diagnose wird eingeschaltet.  
N        : Diagnose wird ausgeschaltet.  
Space   : Beim nächsten CISCON - Aufruf wird SW14 getestet.  
?        : Der Wert wird ausgegeben.

Quittungen:

```
NEW VALUE IS: x  
VALUE IS: x  
ERROR IN PARAMETER FOR CISDIA
```

### 5.3.3 CCLST D Auflistung der DBH - Tabelle

```
$D CCLST D[,p]
```

Auflistung der DBH - Tabelle ab Position p, oder Anfang.

Folgendes wird aufgelistet:

```
TABLE OF DBHS -----  
POS. HOSTNAME ID.  MODE  
nnnn hhhhhhhh  i   mmmmmmmm  
.  
.  
.  
END (xxxx)
```

mit:

nnnn	Position (ab 0001 numeriert)
hhhhhhh	Hostname oder * wenn eigener Host.
i	DBH - id
mmmmmmm	Modus: DBH= DIST= PARTNER=
xxxxx	Anzahl Einträge in der Tabelle.

Beispiel:

```
$D CCLST D  
  
TABLE OF DBHS %%%%%%%%%%  
POS. HOSTNAME ID.  MODE  
0001 *          A   DBH=  
END (0001)
```

### 5.3.4 CCLST P Auflistung der Paßworttabelle

```
$D CCLST P[,p]
```

Auflistung der Paßworttabelle ab Position p, oder Anfang.

Folgendes wird aufgelistet:

```
TABLE OF CIS-PWS-----
POS. CIS-PW DBH
nnnn pppppp hhhhhhhh/i
.
.
.
END (xxxx)
```

mit:

nnnn	Position (ab 0001)
pppppp	CIS - Paßwort
hhhhhhh	Hostname oder 8 Sterne = eigener Host.
i	DBH - id
xxxx	Anzahl Einträge in der Tabelle.

Beispiel (bei DBH=...)

```
$D CCLST P

TABLE OF CIS - PWS
POS. CIS - PW DBH
END (0000)
```

### 5.3.5 CCLST T Auflistung der Transaktionstabelle

```
$D CCLST T[,p]
```

1) Bei Betrieb = N oder U wird folgendes aufgelistet:

```
TABLE OF TRANSACTIONS -----  
POS. TRANS-ID CIS-PW  FLAGS TR.A-DBH          ACT.- DBH  
nnnn tttttttt pppppp X'ff' hhhhhhhh/i      hhhhhhhh/i  
.  
.  
.  
END (xxxx)
```

mit:

nnnn	Position (ab 0001 numeriert)
tttttttt	Transaktionskennung: Betrieb = N TSN + 4 Spaces Betrieb = U KCLOGTER/KCBENID
pppppp	Paßwort - Aktuell
ff	Flags: 01 Transaktion offen 00 Transaktion nicht offen
hhhhhhh	Hostname oder * wenn eigener Host.
i	DBH - id
xxxx	Anzahl Einträge in der Tabelle.

2) Bei Betrieb = S wird folgendes aufgelistet:

```
CONTENT OF TAM-----...-----
STATI  HOSTNAME APPLINAM COUNTER  USER      T PW      PW-TRA
sssssss hhhhhhhh aaaaaaaa cccccc... uuuuuuuu t pppppp pppppp
```

mit:

ssssssss 4 Bytes am Anfang von TAM:

TAMSDBTA	DS	X		Staus of DB - transaction.
TAMTAOPN	EQU	X'80'		- DB TA is open.
TAMTACLI	EQU	X'40'		- Close issued.
TAMTACL	EQU	X'20'		- DB TA closed
TAMTACAN	EQU	X'10'		- DB TA cancelled (S.TAMRBR).
TAMTAUPD	EQU	X'08'		- Update TA.
TAMUTMCO	DS	X		UTM transaction control.
TAMUCLRQ	EQU	X'80'		- User close required.
TAMMULTI	EQU	X'40'		- Multi step transaction allowed.
TAMKPTSK	EQU	X'20'		- Keep task for multi step.
TAMATSKC	EQU	X'10'		- Announce task changes.
TAMRBR	DS	X		Rollback reason.
TAMRBUSR	EQU	X'80'		- User call.
TAMRBADM	EQU	X'40'		- DB administrator.
TAMRBDLK	EQU	X'20'		- Deadlock
TAMRBINT	EQU	X'10'		- Internal error
TAMRBDL	EQU	X'08'		- Task - deadlock.
TAMRBWRM	EQU	X'04'		- Warmstart
TAMTBUER	EQU	X'02'		- User - error.
	DS	X		frei

hhhhhhhh Hostname

aaaaaaaa Application - Name

cccccc... 8 Bytes diverse counter:

TAMSESCN	DS	AL1	Session counter
TAMVGCNT	DS	AL3	Vorgang counter
TAMTACNT	DS	H	Transaction counter
TAMVGNR	DS	H	Vorgang number

uuuuuuuu user (KCLOGTER/KCBENID)

t Typ L = KCLOGTER B = KCBENID

pppppp Paßwort

### 5.3.6 CCMEM Speicherbelegung

```
$D CCMEM[ ,DB.XXXXXX]
```

Antwort am Terminal:

Die ersten 4 Zeilen bringen allgemeine Werte in KB. Ab der 5. Zeile folgt eine Liste von belegten und unbelegten Seiten (4KB - Seiten).

U: . . . . . 0 / . . . . . 242 bedeutet:

```
U      : "Used" (=belegt)
      0 : 1. belegte Seite (dezimal)
     180 : Anzahl belegter Seiten (dezimal)
          = 720 KB
```

F: . . . . . 180 / . . . . . 14 bedeutet:

```
F      : "Free" (=frei)
     180 : 1. freie Seite (dezimal)
     14  : Anzahl freier Seiten (dezimal)
          = 56 KB
```

U: . . . . . 194 / xxxxxxxx usw.

### 5.3.7 CCSYS BS2000 Kommando

```
$D CCSYS /Kommando
```

Das BS2000 - Kommando /Kommando wird mit dem CMD - Makro ausgeführt. Die Ausgabe aus SYSOUT wird auch im Programm empfangen und am Terminal ausgegeben. Es werden maximal 2.000 Bytes abgespeichert. Soll mehr ausgegeben werden, so wird der Text TRUNCATED angeführt.

### 5.3.8 CCT= Angabe einer TSN

```
$D CCT=tttt ,Kommando
```

Das Kommando "Kommando" wird nur ausgeführt, wenn der Task, der es bearbeiten soll, die TSN tttt hat. Betrifft das Kommando "Kommando" das Modul CISCON, so wird bei einer falschen TSN der Fehler CC38 gemeldet. Betrifft das Kommando nicht CISCON, so wird der komplette String an CISDBH weitergegeben und geprüft.

## 5.4 Kommandos für CISUTM

### 5.4.1 Übersicht

\$D UINFT	Info über Transaktionen
\$D UDDIA	Dump der DIAGAEREA
\$D UDUMP	Dump des Programms
\$D UERR	Fehlerabfangroutine
\$D UTERMD	Dump und Programmende
\$D USYS /Kommando	BS - 2000 Kommando
\$D UT=tttt ,Kommando	Angabe einer TSN.

#### 5.4.2 UINFT Info über Transaktionen

```
$D UINFT
```

Auflistung der Transaktionen, die CISUTM bekannt sind. Die Tabelle, die als Eingabe hier dient, ist die, die in CISDBH bei der Funktion DINFT benützt wird.

Folgendes wird aufgelistet:

```
TABLE OF TRANSACTIONS IN CISUTM
POS. HOST      APPLI  TRANS      USER      TSN  M  ADDR.
  1 *          CIS    APP0       APP0       1341 U  A0043
  2 *          CIS    WFM2055M  WFM2055M  1341 U  A0048
  3 *          CIS    TFM1374M  TFM1374M  1341 U  A004D
  4 *          CIS    TFM1375M  TFM1375M  1341 U  A0052
END  (0200)
```

#### 5.4.3 UDDIA Dump der DIAGAREA

```
$D UDDIA
```

Die DIAGAREA wird mit dem Kommando /DUMP auf SYSLST ausgegeben. Die anderen Tasks, die für diese Anwendung arbeiten, geben den Dump der DIAGAREA beim nächsten Aufruf von CISUTM aus.

Dem Aufrufer wird folgendes gemeldet:

```
DUMP OF DIAGAREA nnn
```

#### 5.4.4 UDUMP Dump des Programms

```
$D UDUMP
```

Der Speicherauszug wird mit dem Makro CDUMP ausgegeben. Es wird folgender 2. Titel generiert:

```
DIAGNOSTIC DUMP nnn
```

Die anderen Tasks, die für diese Anwendung arbeiten, geben den Speicherauszug beim nächsten Aufruf von CISUTM raus. Dem Aufrufer wird folgendes gemeldet:

```
DIAGNOSTIC DUMP nnn
```

**5.4.5 UERR Fehlerabfangeroutine**

```
$D UERR=L/eeee , t
```

eeee : CIS - CM, bei der die Aktion t stattfinden soll.

t : Aktion wenn CIS - CM = eeee.  
 E: Dump dieser Task und weiter.  
 D: Dump aller Tasks und weiter.  
 T: Dump aller Tasks und beenden.

L : Gespeicherte Werte auflisten.

**5.4.6 UTERMD Dump und Programmende**

```
$D UTERMD
```

Nach Ausgabe des Speicherauszugs mit dem Makro CDUMP wird das Programm beendet. Die anderen Tasks, die für diese Anwendung arbeiten, geben den Speicherauszug beim nächsten Aufruf von CISUTM aus und beenden sich.

**5.4.7 USYS BS2000 Kommando**

```
$D USYS /Kommando
```

Das BS2000 - Kommando /Kommando wird mit dem CMD - Makro ausgeführt. Die Ausgabe aus SYSOUT wird auch im Programm empfangen und am Terminal ausgegeben. Es werden maximal 2.000 Bytes abgespeichert. Soll mehr ausgegeben werden, so wird der Text TRUNCATED angefügt.

**5.4.8 UT= Angabe einer TSN**

```
$D=tttt ,Kommando
```

Das Kommando "Kommando" wird nur ausgeführt, wenn der Task, der es bearbeiten soll, die TSN tttt hat. Im anderen Fall wird die Meldung UT24 ausgegeben. Das auszuführende Kommando kann irgendein CIS - Kommando sein.

## 5.5 Kommandos für CISKOR

### 5.5.1 Übersicht

Kommando	Erläuterung
\$D KT=tttt,\$D...	Eingabe einer TSN.
/INTR [tsn,]KANC p \$D KANC tttt[,DB.XXXXXX]	Löschen eines Partnereintrags
/INTR [tsn,]KCANF ID=i/NOID, FILE=*ALL/name  \$D KCANF ID=i/NOID,FILE= *ALL/name[,DB.XXXXXX]	Löschen von Dateien in Pool - 3
/INTR [tsn,]KCANS p \$D KCANS p[,DB.XXXXXX]	Löschen einer Statusinformation
/INTR [tsn,]KCANT p \$D KCANT p[,DB.XXXXXX]	Rücksetzen einer Transaktion
/INTR [tsn,]KCRASH \$D KCRASH[,DB.XXXXXX]	Simulierter Crash
/INTR [tsn,]KDOFF \$D KDOFF[,DB.XXXXXX]	Diagnose ausschalten
/INTR [tsn,]KDON \$D KDON[,DB.XXXXXX]	Diagnose einschalten
/INTR [tsn,]KDUMP \$D KDUMP[,DB.XXXXXX]	Ausgeben Dump
/INTR [tsn,]KECHO ttt... \$D KECHO ttt...[,DB.XXXXXX]	Echo - Funktion
/INTR [tsn,]KENA x \$D KENA x[,DB.XXXXXX]	Behandlung der ENA - Comm.
/INTR [tsn,]KEND b \$D KEND b[,DB.XXXXXX]	Setzen der Beendigungsbedingung.
/INTR [tsn,]KINFB[ p] \$D KINFB[ p][,DB.XXXXXX]	Information über Before - Images
/INTR [tsn,]KINFC[ p] \$D KINFC[ p][,DB.XXXXXX]	Information über Partner
/INTR [tsn,]KINFF[...][ p] \$D KINFF[...][ p][,DB.XXXXXX]	Information über Dateien in Pool - 3
/INTR [tsn,]KINFL[ F=dateiname][ p]  \$D KINFL[ F=dateiname] [ p][,DB.XXXXXX]	Information über Sperren.
/INTR [tsn,]KINFL K[z][ p] \$D KINFL K[z][ p][,DB.XXXXXX]	Information über gesperrte Keys
/INTR [tsn,]KINFP[ p] \$D KINFP[ p][,DB.XXXXXX]	Information über Transaktion in PTC
/INTR [tsn,]KINFS[ p] \$D KINFS[ p][,DB.XXXXXX]	Information über Stati
/INTR [tsn,]KINFT[ p] \$D KINFT[ p][,DB.XXXXXX]	Information über Transaktionen

Kommando	Erläuterung
/INTR [tsn,]KKILL \$D KKILL[,DB.XXXXXX]	Ausgeben Dump und Beenden des CISKOOR und aller übergeordneten Tasks
/INTR [tsn,]KMEM \$D KMEM[,DB.XXXXXX]	Speicherplatzbelegung
/INTR [tsn,]KNTASKS n \$D KNTASKS n[,DB.XXXXXX]	Anzahl Tasks vorgeben
/INTR [tsn,]KPEND p \$D KPEND p[,DB.XXXXXX]	PTC beenden
/INTR [tsn,]KPRESET p \$D KPRESET p[,DB.XXXXXX]	PTC rücksetzen
/INTR [tsn,]KSTAT J/Y/N \$D KSTAT J/Y/N	Ausgabe der Statistik am Programmende
/INTR [tsn,]KSYS /cccc \$D KSYS /cccc[,DB.XXXXXX]	BS2000 - Kommando in CISKOOR ausführen.
/INTR [tsn,]KTERMD \$D KTERMD[,DB.XXXXXX]	Ausgeben Dump und Beenden des CISKOOR
/INTR [tsn,]KTIMEOFF \$D KTIMEOFF[,DB.XXXXXX]	TIME ausschalten
/INTR [tsn,]KTIMEON \$D KTIMEON[,DB.XXXXXX]	TIME einschalten
/INTR [tsn,]KTROFF \$D KTROFF[,DB.XXXXXX]	SVC - Trace ausschalten
/INTR [tsn,]KTRON \$D KTRON[,DB.XXXXXX]	SVC - Trace einschalten
/INTR [tsn,]KWAIT n \$D KWAIT n[,DB.XXXXXX]	Warten
/INTR [tsn,]KWRITE \$D KWRITE[,DB.XXXXXX]	CISKOOR überschreibbar machen

### 5.5.2 KT=      Angabe einer TSN

```
$D KT=tttt,$D ...
```

\$D            Operation für "Diagnose".

KT=            Operationsergänzung für Auswahl einer TSN.

tttt            TSN, die den nachfolgenden \$D - Befehl bearbeiten soll. Hat CISKOOR nicht diese TSN, so wird der folgende Befehl nicht ausgeführt. Es erscheint die Meldung:

CISKOOR - NO ACTION (WRONG TSN)

### 5.5.3 KCANC      Löschen eines Partnereintrags

```
$D KCANC p[ ,DB.XXXXXX ]
```

\$D            Operation für "Diagnose".

KCANC            Operationsergänzung für Löschen. Es wird nur der Eintrag in der Partnertabelle gelöscht. Hat der Partner offene Transaktionen, so muß vor KCANC das Kommando KCANT gegeben werden.

p                Position (siehe KINFC)

DB .xxxxxxx    Die DB - Angabe ist nicht notwendig. Wird DB.xxxxxx trotzdem angegeben, wirkt sich das erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Eintrag wurde gelöscht.
CISKOOR - NO ACTION	TSN nicht gefunden.
CISKOOR - ERROR	Eingabe falscher Werte.

### 5.5.4 KCANF Löschen von Dateien in Pool 3

`$D KCANF ID=i/NOID,FILE=*ALL/name[ ,DB.XXXXXX ]`

\$D	Operation für "Diagnose".
KCANF	Operationsergänzung für Löschen Datei(en). Es sollen Dateien im Pool - 3, der allen CISKOORs auf dieser Anlage gemeinsam ist, gelöscht werden.
ID=i	Es sollen Dateien der KOORID=i gelöscht werden.
NOID	Es sollen Dateien des CISKOORs ohne KOORID gelöscht werden.
*ALL	Es sollen alle Dateien dieser ID gelöscht werden.
name	Die Datei mit diesem Namen soll gelöscht werden. Der Name muß vollqualifiziert angegeben werden.
DB .xxxxxx	Die DB - Angabe ist nicht notwendig. Wird DB.xxxxxx trotzdem angegeben, wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Es wurde gelöscht.
CISKOOR - ERROR	Eingabe falscher Werte.

### 5.5.5 KCANS Löschen einer Statusinformation

`$D KCANS p[ ,DB.XXXXXX ]`

\$D	Operation für "Diagnose".
KCANS	Operationsergänzung für Löschen Status.
p	Position der Transaktion (siehe KINFS)
DB .xxxxxx	Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Eintrag wurde gelöscht.
CISKOOR - NO ACTION	Transaktion nicht gefunden.
CISKOOR - ERROR	Eingabe falscher Werte.

### 5.5.6 KCANT Rücksetzen einer Transaktion

```
$D KCANT p [ ,DB.XXXXXXX ]
```

\$D Operation für "Diagnose".

KCANT Operationsergänzung für Rücksetzen

p Position der Transaktion (siehe KINFT)

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Eintrag wurde gelöscht.
CISKOOR - NO ACTION	Transaktion nicht gefunden.
CISKOOR - ERROR	Eingabe falscher Werte.

### 5.5.7 KCRASH Simulation eines Crashes

```
$D KCRASH [ ,DB.XXXXXXX ]
```

\$D Operation für "Diagnose".

KCRASH Operationsergänzung für Simulation eines Crashes. Der CISKOOR - Tasks, der den Befehl erhält, beendet sich, ohne die Datensicherungsdateien zu bearbeiten. Die BIM - Datei ist zum Beispiel offen und kann noch BIM - Sätze enthalten. Die anderen CISKOOR - Tasks beenden sich ebenfalls ohne die Datensicherungsdateien anzufassen.

Dieser Befehl ist nur interessant beim Test des Crash - Verhaltens von CISKOOR.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

CISKOOR - O.K.	Kommando wurde ausgeführt.
----------------	----------------------------

### 5.5.8 KDOFF Diagnose ausschalten

```
$D KDOFF [ ,DB .XXXXXXX ]
```

\$D Operation für "Diagnose".

KDOFF Operationsergänzung für Diagnose ausschalten.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

CISKOOR - O.K.

### 5.5.9 KDON Diagnose einschalten

```
$D KDON [ ,DB .XXXXXXX ]
```

\$D Operation für "Diagnose".

KDON Operationsergänzung für Diagnose einschalten.

Falls eine After - Image - Datei vorhanden ist, werden in diese folgende Sätze geschrieben:

- Before - Image - Sätze
- Sperr- und Entsperr - Sätze
- Status - Abfrage/Stati löschen

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

CISKOOR - O.K.

### 5.5.10 KDUMP Dump von CISKOOR

```
$D KDUMP [ ,DB.XXXXXXX ]
```

\$D Operation für "Diagnose".

KDUMP Operationsergänzung für Dump. Nach Ausgabe eines Dumps wird das Programm fortgesetzt. Der Dump muß mit SODUMP aufbereitet werden.

Der Dump erhält als 2. Überschrift: DIAGNOSTIC DUMP: CISKOOR

im SYSOUT - Protokoll erscheint der Code CISKOOR.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

CISKOOR - O.K. Kommando wurde ausgeführt.

### 5.5.11 KECHO Echo - Funktion

```
$D KECHO ttt... [ ,DB.XXXXXXX ]
```

\$D Operation für "Diagnose".

KECHO Operationsergänzung für "Echo".

ttt... Text

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

KECHO tt...

### 5.5.12 KENA Steuerung von ENA - COMM

\$D KENA x[ ,DB.XXXXXXX ]

\$D Operation für "Diagnose".

KENA Operationsergänzung für Steuerung von ENA - COMM.

x Aktion: - L (Lock) : ENA - COMM sperren.  
 - U (Unlock) : ENA - COMM wieder freigeben.  
 - D (Display) : den aktuellen Zustand ausgeben.

Wird ENA - COMM gesperrt, so kann sich kein neuer Partner an CISKOOR anschliessen.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

CISKOOR - O.K. Kommando wurde ausgeführt.

CISKOOR - STATE: xxxx Anforderung der Auflistung.

mögliche Werte: OK : alles O.K.  
 OKLE : O.K. aber ENA gesperrt.  
 ENDE : Session - beendet.  
 LADE : CISKOOR wird geladen.  
 RST : Restart läuft.  
 FRST : Fehler bei Restart.  
 KILL : Kill - Kommando gegeben.

### 5.5.13 KEND      Setzen der Beendigungsbedingung

```
$D KEND b[ ,DB.XXXXXXX ]
```

\$D            Operation für "Diagnose".

KEND        Operationsergänzung für Beendigungsbedingung.

b            Beendigungsbedingung: 1 = CISKOOR wird beendet, wenn sich der letzte Partner abmeldet.

2 = CISKOOR beendet sich nicht zwischen der "Von"- und der "Bis" - Uhrzeit. Außerhalb dieser Uhrzeiten geht CISKOOR auf Bedingung 1 und beendet sich, wenn sich der letzte Partner abmeldet.

3 = CISKOOR beendet sich nicht. Zur Beendigung muß mit einem nachfolgenden Kommando die Bedingung auf 1 oder 2 geändert werden.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.

Beendigungsbedingung geändert.

CISKOOR - ERROR

Eingabe einer falschen Beendigungsbedingung.



### 5.5.15 KINFC Information über Partner

```
$D KINFC[ p][ ,DB.XXXXXX]
```

ab V11

```
$D INFC[ p][ ,DB.XXXXXX]
```

(altes Format)

\$D Operation für "Diagnose".

KINFC/INFC Operationsergänzung für Information über Partner.

p Position, ab der protokolliert wird. Wird p nicht angegeben, wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.

DB.xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-COMM. *****JJJJN 1 0700 1800*003/001***172243
*POS *TSN USER-ID. **MP-ADR SIZE(4K) MODE **TIME
      5046 VD5DB      00020000
      1 5044 VD5DB      00021000 00000009 N 11      172111
END (100)
```

Erläuterung der Überschriftzeile:

JJJJN "Vorhanden" - Kennzeichen für: After - Image - Datei : J  
Before - Image - Datei : J  
Protokoll - Datei : J  
Transaktionen : J  
Diagnose - Ein : N

1 CISKOOR - Beendigungsbedingung

0700 "Von" - Uhrzeit

1800 "Bis" - Uhrzeit

003/001 Max. Anzahl / aktive Anzahl CISKOOR - Tasks.

172243 Uhrzeit der Protokollierung (Stunde, Minuten, Sekunden).

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Partner. Die 1. Zeile betrifft immer CISKOOR selbst und hat keine Positionsnummer.  Nach der letzten Datenzeile erscheint: END (xxx). xxx = max. Anzahl der angeschlossenen Partner. (siehe auch Parameter KNCOMM).
TSN	Task - Sequence - Number des Partners (bzw. von CISKOOR in der ersten Zeile).
USER-ID	Benutzerkennung des Partners (bzw. von CISKOOR in der 1. Zeile).
MP-ADR	Adresse des Bereichs für diesen Partner im Common - Memory - Pool.
SIZE	Größe des Bereichs im Common - Memory Pool - in Vielfachem von 4K.
MODE	Betriebsart des Partners: N = Teilnehmer (TIAM) U = UTM (nicht synchronisiert) S = UTM - synchronisiert  es folgt die Version des Partners (9,10,11)
TIME	Uhrzeit der Anmeldung des Partners (Stunde, Minuten, Sekunden)

**5.5.16 KINFF      Information über Dateien in Pool 3**

```
$D KINFF[ ALL/ID=i/ID=*/NOID][ p][ ,DB.XXXXXX]
```

- \$D                    Operation für "Diagnose".
- KINFF                Operationsergänzung für Informationen über Dateien in Pool - 3. Hier steht in einem gemeinsamen Pool welcher CISKOOR welche Dateien bearbeitet. Beendet sich ein CISKOOR, so werden "seine" Dateien ausgegeben. CISKOOR steht immer für einen Taskverbund.
- ALL                    Es soll eine Übersicht der Anzahl Dateien pro KOORID ausgegeben werden. Diese Informationen passen immer auf einen Bildschirm. Eine Position braucht nicht angegeben zu werden.
- ID=i                    Es sollen die Dateien für die KOORID i ausgegeben werden. Fehlt die Angabe ID=i und NOID, so wird die eigene KOORID genommen.
- ID=\*                    Es sollen die Dateien für alle KOORIDs ausgegeben werden.
- NOID                    Es sollen die Dateien für CISKOOR ohne KOORID ausgegeben werden.
- p                        Position, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.
- DB .xxxxxxx            Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung mit ALL:

```
**** INFO-FILE ** ALL *****
*****
ID #FILES ID #FILES ID #FILES ID #FILES ID ... #FILES ID
#FILES
NO      2  A      6  B      4  N      8      ...
END
```

Erläuterung der Datenzeilen:

- ID                    KOORID oder "NO" wenn ohne KOORID.
- #FILES                Anzahl Dateien für diese KOORID.

Beispiel einer Auflistung mit ID=\*:

```

**** INFO-FILE ** ID=* *****
****POS ID FILENAME
      1 NO :C:$UID.HD.PASS01
      2 NO :C:$UID.VD.PASS01
      3 A  :C:$UID.HD.PASS02
      4 A  :C:$UID.VD.PASS02
END

```

Erläuterung der Datenzeilen:

POS	Position in der Tabelle der Dateien in POOL - 3. Die neuen Dateien werden immer ans Ende der schon existierenden Tabelle angefügt. Wird ein CISKOR beendet, dann werden die Dateien seiner KOORID entfernt. Die restlichen Einträge werden jeweils nach links verschoben, so daß die Tabelle wieder lückenlos durchnummeriert ist.
ID	KOORID oder "NO" wenn ohne KOORID.
FILENAME	Vollqualifizierter Name der Datei.





### 5.5.18 KINFP Information über Transaktionen in PTC

```
$D KINFP [ p ] [ ,DB.XXXXXX ]
```

- \$D Operation für "Diagnose".
- KINFP Operationsergänzung für Information über Transaktionen im Zustand PTC.
- p Position, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.
- DB.xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-PETA-CISKOOR
*****
*POS FLAG HOSTNAME APPLINAM *****TID **ID
  1 NU * 4711 0001
===> INFO-TRANS-CISKOOR
END (400)
```

Erläuterung der Datenzeilen:

- POS Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint:END (xxx). (xxx) = Anzahl Einträge in der Tabelle. (siehe Parameter KNSTA).
  - FLAG Diverse Kennungen (4 Bytes):
    - 1. Byte: Betriebsart (S, U, N)
    - 2. Byte: Herkunft (zur Zeit immer U)
    - 3. Byte: leer
    - 4. Byte: leer
  - HOSTNAME Name des Prozessors.
  - APPLINAM Name der Anwendung.
  - TID Transaktionskennung
  - ID Inter Kennung
- ===> INFO-TRANS-CISKOOR

bedeutet, daß es Transaktionen im Zustand PTC gibt, die in der Transaktionstabelle enthalten sind. (Kommando KINFT).

### 5.5.19 KINFS Information über Stati

```
$D KINFS[ p][ ,DB.XXXXXX]
```

- \$D            Operation für "Diagnose".
- KINFS        Operationsergänzung für Information über Stati.
- p            Position, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.
- DB .xxxxxx   Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-STATI-CISKOOR
*****
*POS FLAG HOSTNAME APPLINAM PAM-PAGE SC #TAS 1ST TID LAST
      TID
  1 NU *           00000002 00      1 4711
END (400)
```

Erläuterung der Datenzeilen:

- POS            Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx). (xxx) = Anzahl Einträge in der Tabelle. (siehe Parameter KNSTA).
- FLAG            Diverse Kennungen (4 Bytes): 1. Byte: Betriebsart (S, U, N)  
2. Byte: Herkunft (zur Zeit immer U)  
3. Byte: leer  
4. Byte: leer
- HOSTNAME        Name des Prozessors.
- APPLINAM        Name der Anwendung.
- PAM-PAGE        Pam - Page in der Statusdatei.
- SC             UTM - Session - Counter.
- #TAS            Anzahl Transaktionen.
- 1ST TID         Erste gespeicherte TID.
- LAST TID        Letzte gespeicherte TID.

### 5.5.20 KINFT Information über Transaktionen

\$D KINFT[ p][ ,DB.XXXXXX]	ab V11
----------------------------	--------

\$D INFT[ p][ ,DB.XXXXXX]	altes Format
---------------------------	--------------

\$D Operation für "Diagnose".

KINFT/INFT Operationsergänzung für Information über Transaktionen.

p Position, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.

DB.xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-TRANS-CISKOOR ****
*POS HOSTNAME APPLINAM **TID ***UTM-TID SP. *TSN MODE *USER
  1 *                5044                P  5044 N  50444
END (100)
```

Erläuterung der Datenzeilen:

- POS Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx). (xxx) = max. Anzahl der aktuellen Transaktionen. (siehe auch Parameter KNTRANS).
- HOSTNAME Name des Rechners in dem die Transaktion ihren Ursprung hat. (\* heißt eigener Rechner).
- APPLINAM Name der Anwendung, in der die Transaktion ihren Ursprung hat. Stehen hier Spaces, so ist der Partner ein Teilnehmerprogramm.
- TID Transaktions - Identifikation: bei Betriebsart N: TSN plus 4 Spaces  
bei Betriebsart U: logischer Terminalname (KCLOGTER)  
bei Betriebsart S: Vorgang - Nummer (TAMVGNR) abdruckbar plus 4 Spaces



### 5.5.21 KKILL      Dump und Ende des gesamten Systems

```
$D KKILL[ ,DB.XXXXXX ]
```

\$D            Operation für "Diagnose".

KKILL        Operationsergänzung für Dump und Ende des Systems. Es wird ein Dump ausgegeben, ein Kennzeichen im Memory - Pool gesetzt und dann CISKOOR beendet. Alle Partner von CISKOOR werden beim nächsten Aufruf auf Grund des Kennzeichens im Memory - Pool mit Dump beendet mit dem Code PARTNER und der 2. Überschrift im Dump:

DIAGNOSTIC DUMP: PARTNER.

Die Dumps müssen per SODUMP aufbereitet werden.

Der CISKOOR - Dump erhält als 2. Überschrift:

DIAGNOSTIC DUMP: CISKOOR

Im SYSOUT - Protokoll erscheint der Code CISKOOR.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:                      CISKOOR - O.K. wenn Kommando ausgeführt wurde.

**5.5.22 KMEM Speicherbelegung**

\$D KMEM[ ,DB .XXXXXX ]
-------------------------

\$D            Operation für "Diagnose".

KMEM        Operationsergänzung für Speicherbelegung.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Antwort am Terminal:

Die ersten 4 Zeilen bringen allgemeine Werte in KB. Ab der 5. Zeile folgt eine Liste von belegten und unbelegten Seiten (4KB - Seiten).

U : . . . . . 0 / . . . . . 242 bedeutet:

U            : "Used" (=belegt)  
           0    : 1. belegte Seite (dezimal)  
           242 : Anzahl belegter Seiten (dezimal)  
               = 968 KB

F : . . . . . 242 / . . . . . 14 bedeutet:

F            : "Free" (=frei)  
           242 : 1. freie Seite (dezimal)  
           14  : Anzahl freier Seiten (dezimal)  
               = 56 KB

U : . . . . . 256 / xxxxxxxx usw.

### 5.5.23 KNTASKS Setzen der Anzahl Tasks

```
$D KNTASKS n[ ,DB .XXXXXX ]
```

\$D Operation für "Diagnose".

KNTASKS Operationsergänzung für Setzen Anzahl Tasks.

n Neue Anzahl der Tasks. Ist n größer als der bisherige Wert, so werden weitere Tasks gestartet. Hierzu werden in CISKOOR /ENTER - Kommandos abgesetzt. Es muß der CISKOOR - Parameter KENTER vorhanden sein. Ist n kleiner als der bisherige Wert, so werden Tasks beendet, bis nur noch die neue Anzahl TASKS tätig ist. Die Tasks beenden sich immer nur nach dem Ende eines Zyklusses, und nie mittendrin. n kann nicht größer als der Maximalwert (KMAXT=) werden, und auch nicht kleiner als 1. Wird einer dieser Werte angegeben, so wird ohne Fehlermeldung max. bzw. 1 angenommen.

DB .xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Kommando ausgeführt.
CISKOOR - ERROR	Fehler bei /ENTER.

### 5.5.24 KPEND PTC beenden

```
$D KPEND p[ ,DB .XXXXXX ]
```

\$D Operation für "Diagnose".

KPEND Operationsergänzung für PTC beenden. Die angegebene Transaktion (siehe p) wird beendet.

p Position der Transaktion beim Aufruf KINFP.

DB .xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Transaktion beendet.
CISKOOR - ERROR	Fehler.

### 5.5.25 KPRESET PTC rücksetzen

```
$D KPRESET p[ ,DB .XXXXXX ]
```

\$D Operation für "Diagnose".

KPRESET Operationsergänzung für PTC rücksetzen. Die angegebene Transaktion (siehe p) wird zurückgesetzt.

p Position der Transaktion beim Aufruf KINFP.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Transaktion rückgesetzt.
CISKOOR - ERROR	Fehler.

### 5.5.26 KSTAT Ausgabe der Statistik ein/ ausschalten

```
$D KSTAT J/Y/N[ ,DB .XXXXXX ]
```

\$D Operation für "Diagnose".

KSTAT Operationsergänzung für Statistik ein - /ausschalten.

J/Y Statistik soll bei Programmende ausgegeben werden.

N Statistik soll nicht ausgegeben werden (Standardfall).

Das Format der Statistikausgabe wird nicht erläutert. Es sind Diagnoseinformationen für die CIS - Beratung.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Kommando ausgeführt.
CISKOOR - ERROR	Eingabe falscher Werte.

### 5.5.27 KSYS      BS2000 Kommando ausführen

```
$D KSYS /ccc[ ,DB .XXXXXX ]
```

\$D            Operation für "Diagnose".

KSYS        Operationsergänzung für BS2000 - Kommando.

/ccc        BS2000 - Kommando (inklusive Schrägstrich). Das Kommando wird über CMD an das Betriebssystem weitergegeben.

In CISKOOR ist ein Antwortbereich von 500 Bytes vorhanden. Gibt das Kommando eine Antwort aus, so wird diese an CIS übergeben und am Terminal ausgegeben.

Der Aufruf:    \$D KSYS /D %10.(S=0,L=24)

             liefert folgende Anfangsadressen:

1. Wort: Adresse von CISVARI
2. Wort: Adresse von CISKOOR
3. Wort: Adresse von CISAIM
4. Wort: Adresse von CISBIM
5. Wort: Adresse von CISPRO
6. Wort: Adresse von CISSP

Der Aufruf:    \$D KSYS /D %MR

             liefert folgende Registerinhalte:

R3, R4, R5, R6, R7	Basis CISKOOR
R8	Adresse VMADL
R9	Basis VCISKOOR
R10	(siehe oben)
R11	Adresse POOL - Anfang

DB .xxxxxxx    Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

Keine (eventuell Protokollierung nach Kommando) alles o.k.

CISKOOR - TRUNCATED	Wenn abgeschnitten wurde.
CISKOOR - ERROR	Fehler.

### 5.5.28 KTERMD Dump und Ende von CISKOOR

```
$D KTERMD[ ,DB .XXXXXX ]
```

\$D Operation für "Diagnose".

KTERMD Operationsergänzung für Dump und Ende. Nach Ausgabe eines Dump wird CISKOOR beendet. Der Dump muß per SODUMP aufbereitet werden.

Der Dump erhält als 2. Überschrift:

DIAGNOSTIC DUMP: CISKOOR

im SYSOUT - Protokoll erscheint der Code CISKOOR.

DB .xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittung am Terminal:

CISKOOR - O.K.                      Kommando ausgeführt.

### 5.5.29 KTIMEOFF TIME ausschalten

```
$D KTIMEOFF[ ,DB .XXXXXX ]
```

\$D Operation für "Diagnose".

KTIMEOFF Operationsergänzung für TIME aus.

DB .xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.                      TIME ausgeschaltet.  
CISKOOR - ERROR                    Fehler.

### 5.5.30 KTIMEON TIME einschalten

```
$D KTIMEON[ ,DB.XXXXXX ]
```

(neues Format)

```
$D KTIMEOLD[ ,DB.XXXXXX ]
```

(altes Format)

\$D Operation für "Diagnose".

KTIMEON Operationsergänzung für TIME ein (neu). Die TIME - Sätze werden in die Datei CIS.TIME.tttt (ttt = TSN) geschrieben. Sie müssen mit dem Programm CISTA ausgewertet werden.

KTIMEOLD Operationsergänzung für TIME ein (alt). Die TIME - Sätze werden direkt nach SYSLS01 geschrieben. Sie können mit dem /PRINT - Kommando ausgedruckt werden. Diese Möglichkeit belastet CISKOOR mehr als die neue Funktion. Auch können weniger Werte ausgegeben werden.

DB .xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	TIME eingeschaltet.
CISKOOR - ERROR	Fehler.

### 5.5.31 KTROFF SVC - Trace ausschalten

```
$D KTROFF[ ,DB.XXXXXX ]
```

\$D Operation für "Diagnose".

KTROFF Operationsergänzung für SVC - Trace aus.

DB .xxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	SVC - Trace ausgeschaltet.
CISKOOR - ERROR	Fehler.

### 5.5.32 KTRON SVC - Trace einschalten

```
$D KTRON[ ,DB.XXXXXXX ]
```

\$D Operation für "Diagnose".

KTRON Operationsergänzung für SVC - Trace ein.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	SVC - Trace eingeschaltet.
CISKOOR - ERROR	Fehler.

### 5.5.33 KWAIT Warten

```
$D KWAIT n[ ,DB.XXXXXXX ]
```

\$D Operation für "Diagnose".

KWAIT Operationsergänzung für Warten.

n Anzahl Sekunden, die zu warten ist.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Kommando ausgeführt.
----------------	----------------------

### 5.5.34 KWRITE CISKOOR überschreibbar machen

```
$D KWRITE[ ,DB.XXXXXXX ]
```

\$D            Operation für "Diagnose".

KWRITE        Operationsergänzung für überschreibbar machen.

DB .xxxxxxx Die DB - Angabe kann entfallen. Wird DB.xxxxxx trotzdem angegeben, so wirkt es sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

CISKOOR - O.K.	Seiten sind überschreibbar.
CISKOOR - ERROR	Fehler.

## 5.6 Kommandos für CISDBH

### 5.6.1 Übersicht

Kommando	Erläuterung
\$D DT=tttt, kommando[ ,DB.xxxxxx ]	Angabe einer TSN.
/INTR [tsn,]DCANC p \$D DCANC p[ ,DB.xxxxxx ]	Löschen eines Partnereintrags.
/INTR [tsn,]DCL, DB.xxxxxx \$D DCL, DB.xxxxxx	Datei(en) zu DB.xxxxxx vorübergehend schließen.
/INTR [tsn,]DCM \$D DCM[ ,DB.xxxxxx ]	CMs der verschiedenen Tasks abfragen.
/INTR [tsn,]DCMCL \$D DCMCL[ ,DB.xxxxxx ]	CMs der verschiedenen Tasks löschen.
/INTR [tsn,]DDUMP \$D DDUMP[ ,DB.xxxxxx ]	Ausgeben Dump in allen DBH - Tasks.
/INTR [tsn,]DENA x \$D DENA x[ ,DB.xxxxxx ]	Behandlung der ENA - Comm.
/INTR [tsn,]DEND b \$D DEND b[ ,DB.xxxxxx ]	Setzen Beendigungsbedingung von CISDBH.
/INTR [tsn,]DINFC[ p] \$D DINFC[ p][ ,DB.xxxxxx ]	Information über Partner.
/INTR [tsn,]DINFT[ p] \$D DINFT[ p][ ,DB.xxxxxx ]	Information über Transaktionen.
/INTR [tsn,]DKILL \$D DKILL[ ,DB.xxxxxx ]	Ausgeben Dump und Beenden aller DBH - Tasks und aller übergeordneten Tasks.
/INTR [tsn,]DNTASKS n \$D DNTASKS n[ ,DB.xxxxxx ]	Setzen der Anzahl Tasks.
/INTR [tsn,]DOP, DB.xxxxxx \$D DOP, DB.xxxxxx	Datei(en) zu DB.xxxxxx wieder eröffnen.
/INTR [tsn,]DSEC, DB.xxxxxx \$D DSEC, DB.xxxxxx	Datei(en) zu DB.xxxxxx vorreservieren.
/INTR [tsn,]DSTAT J/Y/N \$D DSTAT J/Y/N[ ,DB.xxxxxx ]	Ausgabe der Statistik am Programmende.
/INTR [tsn,]DTERMD \$D DTERMD[ ,DB.xxxxxx ]	Ausgeben Dump und Beenden aller DBH - Tasks.
/INTR [tsn,]DUNSEC \$D DUNSEC[ ,DB.xxxxxx ]	Vorreservierte Dateien dieser Transaktion wieder freigeben.
/INTR [tsn,]DWAIT n \$D DWAIT n[ ,DB.xxxxxx ]	Warten

### 5.6.2 DT=      Angabe einer TSN

```
$D DT=tttt ,kommando[ ,DB.xxxxxx ]
```

\$D            Operation für "Diagnose".

DT            Operationsergänzung für TSN.

tttt          TSN des CISDBH.

kommando    CIS - Kommando. Es muß ein \$D - Kommando sein. Dieses Kommando wird nur ausgeführt, wenn es im CISDBH mit der angegebenen TSN "angekommen" ist.

xxxxxx      Datenbeschreibung

Quittungen am Terminal:

              Siehe kommando

### 5.6.3 DCANC      Löschen eines Partnereintrags

```
$D DCANC p[ ,DB.xxxxxx ]
```

\$D            Operation für "Diagnose".

DCANC        Operationsergänzung für Löschen.

p             Position in der Partnertabelle (Siehe DINFC).

xxxxxx      Datenbeschreibung

Quittungen am Terminal:

CISDBH - O.K.

Eintrag gelöscht.

CISDBH - ERROR

Eingabe einer falschen Position.

### 5.6.4 DCL Datei(en) vorübergehend schließen

\$D DCL ,DB .xxxxxxx

\$D Operation für "Diagnose"

DCL Operationsergänzung für Datei(en) schließen (CLOSE). Dieses Kommando geht an einen der CISDBH - Tasks (bzw. an den einzigen CISDBH - Task).

Die HD wird in der CISKOOR - Sperrverwaltung reserviert damit in dieser Datei keine Transaktion mehr etwas sperren kann. Wenn noch etwas gesperrt ist, wird die Operation mit SP04 abgebrochen. Mit dem Kommando \$D DSEC,DB.xxxxxx kann sichergestellt werden, daß neue Transaktionen in dieser Datei nichts sperren können.

Die HD (und die VD) wird in diesem CISDBH - Task geschlossen.

Die anderen CISDBH - Tasks werden aufgerufen, die Datei(en) zu schließen. Jeder Task gibt eine positive oder negative Rückmeldung. Es wird jedoch nur sehr kurz auf Antworten gewartet. Sie können jederzeit mit \$D DCM abgefragt werden.

DB .xxxxxxx Das Datenbankpaßwort muß angegeben werden, um die HD und die eventuell vorhandene VD bekannt zu machen.

Quittungen am Terminal:

CISDBH - TSN: tttt - O.K.	Alles O.K. im Task tttt.
CISDBH - TSN: tttt - NO CM	Noch keine Antwort.
CISDBH - TSN: tttt - ERROR: eeee	Fehler eeee im Task tttt.

### 5.6.5 DCM CM der Tasks abfragen

\$D DCM[ ,DB .xxxxxxx ]

\$D Operation für "Diagnose".

DCM Operationsergänzung für CM abfragen. Dieses Kommando geht an einen der CISDBH - Tasks (bzw. an den CISDBH - Task) und bringt eine Liste aller TSNs und der zugehörigen CMs des vorhergehenden DCL bzw. DOP - Kommandos. Es wird benützt, wenn ein Task längere Zeit arbeitet und die Datei(en) nicht sofort schließen oder öffnen kann.

xxxxxxx Datenbeschreibung

Quittungen am Terminal:

CISDBH - TSN: tttt - O.K.	Alles O.K. im Task tttt.
CISDBH - TSN: tttt - NO CM	Noch keine Antwort.
CISDBH - TSN: tttt - ERROR: eeee	Fehler eeee im Task tttt.

### 5.6.6 DCMCL      CM der Tasks löschen

```
$D DCMCL [ ,DB .xxxxxxx ]
```

\$D                    Operation für "Diagnose".

DCMCL                Operationsergänzung für CM löschen (CLEAR). Dieses Kommando geht an einen der CISDBH - Tasks (bzw. an den CISDBH - Task) und löscht die CMs aller Tasks.

xxxxxxx              Datenbeschreibung

Quittung am Terminal:

CISDBH - o.k.

### 5.6.7 DDUMP      Dump aller DBH - Tasks

```
$D DDUMP [ ,DB .xxxxxxx ]
```

\$D                    Operation für "Diagnose".

DDUMP                Operationsergänzung für Dump. Es wird in allen Tasks dieses DBHs ein Dump ausgegeben. Nach Ausgabe laufen die Programme normal weiter. Die Dumps müssen mit SODUMP aufbereitet werden.

Das Kommando wird in einem der DBH - Tasks bearbeitet. Der Dump erhält als 2. Überschrift:

DIAGNOSTIC DUMP: CISDBH1

Im SYSOUT - Protokoll erscheint der Code CISDBH1. Die anderen DBH - Tasks erhalten ein Signal, das eine Contingency - Routine veranlaßt zu dumpen mit dem Code CISDBH2 und der 2. Überschrift:

DIAGNOSTIC DUMP: CISDBH2.

xxxxxxx              Datenbeschreibung

Quittung am Terminal:

CISDBH - O.K.                    Kommando wurde ausgeführt.

### 5.6.8 DENA Steuerung von ENA - COMM

\$D DENA x[ ,DB .xxxxxxx ]

\$D Operation für "Diagnose".

DENA Operationsergänzung für Steuerung von ENA - COMM.

x Aktion: - L (Lock) : ENA - COMM sperren.  
 - U (Unlock) : ENA - COMM wieder freigeben.  
 - D (Display) : Den aktuellen Zustand ausgeben.

Wird ENA - COMM gesperrt, so kann sich kein neuer Partner an CISDBH anschliessen.

xxxxxxx Datenbeschreibung

Quittung am Terminal:

CISDBH - O.K. Kommando ausgeführt.  
 CISDBH - STATE: xxxx Bei Anforderung der Auflistung.  
 Mögliche Werte: OK : alles O.K.  
 OKLE : O.K. aber ENA gesperrt.  
 ENDE : Session - beendet.  
 LADE : CISDBH wird geladen.  
 KILL : Kill - Kommando gegeben.

### 5.6.9 DEND Setzen der Beendigungsbedingung von CISDBH

\$D DEND b[ ,DB .xxxxxxx ]

\$D Operation für "Diagnose".

DEND Operationsergänzung für Beendigungsbedingung.

b Beendigungsbedingung: 1 = CISDBH wird beendet, wenn der letzte Partner sich abmeldet.

2 = CISDBH beendet sich nicht zwischen der "Von"- und der "Bis" - Uhrzeit. Außerhalb dieser Uhrzeiten geht CISDBH auf Bedingung 1 und beendet sich, wenn der letzte Partner sich abmeldet.

3 = CISDBH beendet sich nicht. Zur Beendigung muß mit einem nachfolgenden Kommando die Bedingung auf 1 oder 2 geändert werden.

xxxxxxx Datenbeschreibung

Quittungen am Terminal: CISDBH - O.K. Beendigungsbedingung geändert.  
 CISDBH - ERROR Falsche Beendigungsbedingung.

### 5.6.10 DINFC Information über Partner

```
$D DINFC[ p][ ,DB .xxxxxxx]
```

- \$D Operation für "Diagnose".
- DINFC Operationsergänzung für Information über Partner.
- p Position, ab der protokolliert wird. Wird p nicht angegeben, wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.
- xxxxxxx Datenbeschreibung

Beispiel einer Auflistung:

```
---- INFO-COMM. --DBHID=A -----2 0700 1800-----010/004-1-164431
-POS -TSN USER-ID.--MP-ADR ----SIZE COMM MODE APPLINAM HOSTNAME --TIME
      3472 VD5DB 000B0000                CISDBHA  HOST1
      3471
      3473
      3474
      3 3494 VD5DB 000C3000 00000009 P   N 11          *          164325
END (100)
```

Erläuterung der Überschriftzeile:

- DBHID=A Kennung des Data - Base - Handlers.
- 2 Beendigungsbedingung
- 0700 "Von" - Uhrzeit
- 1800 "Bis" - Uhrzeit
- 010/004-1 Es können maximal 10 Tasks vorhanden sein, zur Zeit gibt es jedoch nur vier. Die Angabe - 1 zeigt, daß von den 4 Tasks einer ein Verwaltungstask ist.
- 164431 Uhrzeit der Protokollierung. (Stunde, Minuten, Sekunden)

Erläuterung der Datenzeilen:

- POS Position in der internen Tabelle der Partner. Die ersten Zeilen betreffen immer den Data - Base - Handler selbst und haben keine Positionsnummern. Nach der letzten Datenzeile erscheint: END (xxx) . xxx = max. Anzahl der angeschlossenen Partner. (siehe auch Parameter DNCOMM)

- TSN** Task - Sequence - Number des Partners (bzw. Data - Base - Handlers in den ersten Zeile). Beim Multitask - Betrieb werden in den ersten Zeilen die TSNs der DBHs aufgelistet. In der 1. Zeile ist immer die TSN des Tasks, der die Information ausgegeben hat. Wenn es einen Verwaltungstask gibt, so steht seine TSN immer in der 2. Zeile.
- USER-ID** Benutzerkennung des Partners (bzw. Data - Base - Handlers in der 1. Zeile).
- MP-ADR** Adresse des Bereichs für diesen Partner im Common - Memory - Pool. Verkehrt der Partner über DCAM (siehe Spalte (COMM), so steht hier die Adresse des Bereichs, in dem die Nachrichten gespeichert werden.
- SIZE** Bei P1 - Eventing (P in Spalte COMM): Größe des Memory - Pool Bereichs in Vielfachem von 4 K.  
Bei DCAM (D in Spalte COMM) 4 Stellen vorgegebene Elementlänge (Konstante in CISCON=X'07D0'=2000).  
4 Stellen maximale Länge für diese Verbindung.
- COMM** Art der Kommunikation: P = über P1 - Eventing und Common - Memory - Pool  
D = über DCAM
- MODE** Betriebsart des Partners: N = Teilnehmer (TIAM)  
S = UTM - Synchronisiert
- Hinter der Betriebsart steht die Version des Partners. (9, 10, 11).
- APPLINAM** Name der Anwendung 1. Zeile (Data Base Handler):  
Leer wenn kein DCAM - Anschluß (DCAM=N).  
CISDBHx wenn DCAM - Anschluß mit x = DBH Kennung.
- Weitere Zeilen (Partner): Der Name der Anwendung. Er hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	APPLINAM
P	N	Spaces
P	S	Name der UTM - Anwendung. Siehe: MAX APPLINAME=... für KDCDEF
D	N	CIS tttt tttt = TSN des Partners
D	S	Name der UTM - Anwendung.

## Spezielle Kommandos

HOSTNAME      Name des Rechners      1. Zeile (Data Base Handler):  
Leer wenn kein DCAM - Anschluß  
Rechnername wenn DCAM - Anschluß.

Weitere Zeilen (Partner): Der Name des Rechners. Er hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	HOSTNAME
P	N	*
P	S	Name des Rechners.
D	alle	

TIME            Uhrzeit des Verbindungsaufbaus (Stunden, Minuten, Sekunden). In der 1. Zeile steht die Zeit des Ladens des DBHs.

**5.6.11 DINFT Information über Transaktionen**

```
$D DINFT[ p][ ,DB .xxxxxx]
```

\$D Operation für "Diagnose".

DINFT Operationsergänzung für Information über Transaktionen.

p Position, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben. Die Position kann einstellig oder mehrstellig angegeben werden.

xxxxxx Datenbeschreibung

Beispiel einer Auflistung:

```
----- INFO-TRANS-CISDBH -----
-POS HOSTNAME APPLINAM ----USER -----TID -TSN MODE -PAGE
  1 *                4687    4687                4687 N   A0042
  2 GZ432050 CISUTM   ADMIN    0100000100010019 4687 S   A0047
END (100)
```

Erläuterung der Datenzeilen:

POS Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx). (xxx) = max. Anzahl der aktuellen Transaktionen. (siehe auch Parameter DNTRANS).

HOSTNAME Name des Rechners. Der Name des Rechners hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	HOSTNAME
P	N	*
P	S	Name des Rechners.
D	alle	

APPLINAM Name der Anwendung. Der Name der Anwendung hängt von der Betriebsart und der Art der Kommunikation ab.

COMM	MODE	APPLINAM
P	N	Spaces
P	S	Name der UTM - Anwendung. Siehe: MAX APPLINAME=... für KDCDEF
D	N	CISttt tt = TSN des Partners
D	S	Name der UTM - Anwendung.

## Spezielle Kommandos

USER

MODE	USER
S	KCBENID/KCLOGTER
sonst.	TSN

TID

Transaktionskennung

MODE	USER
S	Interne UTM - Transaktionskennung
sonst.	TSN des Partners

TSN

Task - Sequence - Number des Partners.

MODE

Betriebsart des Partners: N = Teilnehmer (TIAM)  
S = UTM - synchronisiert

PAGE

Interne Nummer. Zeigt wo die Daten dieser Transaktion gespeichert werden:

Aaaaa: Nr. der Seite im Memory - Pool  
Ppppp: Nr. der PAM - Page in der Datei  
CISUTM.RIOFILE.CISDBHi

### 5.6.12 DKILL Dump und Ende des gesamten Systems

```
$D DKILL[ ,DB .xxxxxxx ]
```

\$D Operation für "Diagnose".

DKILL Operationsergänzung für Dump und Ende des Systems.

Es wird in allen Tasks dieses DBHs ein Dump ausgegeben. Nach Ausgabe wird im Memory - Pool ein Kennzeichen gesetzt und die Programme werden beendet. Alle Partner des DBH werden beim nächsten Aufruf auf Grund des Kennzeichens im Memory - Pool mit Dump beendet mit dem Code PARTNER und der 2. Überschrift im Dump:

DIAGNOSTIC DUMP: PARTNER.

Die Dumps müssen mit SODUMP aufbereitet werden.

Das Kommando wird in einem der DBH - Tasks bearbeitet. Hier erhält der Dump als 2. Überschrift:

DIAGNOSTIC DUMP: CISDBH1

im SYSOUT - Protokoll erscheint der Code CISDBH1. Die anderen DBH - Tasks erhalten ein Signal, das eine Contingency - Routine veranlaßt zu dumpen mit dem Code CISDBH2 und der 2. Überschrift:

DIAGNOSTIC DUMP: CISDBH2.

xxxxxxx Datenbeschreibung

Quittung am Terminal:

CISDBH - O.K. Kommando ausgeführt.

### 5.6.13 DNTASKS Setzen der Anzahl Tasks

```
$D DNTASKS n[ ,DB .xxxxxxx ]
```

\$D            Operation für "Diagnose".

DNTASKS     Operationsergänzung für Setzen Anzahl Tasks.

n            Neue Anzahl der Tasks.

Ist n größer als der bisherige Wert, so werden weitere Tasks gestartet. Hierzu werden in CISDBH /ENTER - Kommandos abgesetzt. Der CISDBH - Parameter DENTER muß vorhanden sein.

Ist n kleiner als der bisherige Wert, so werden Tasks beendet, bis nur noch die neue Anzahl TASKS aktiv ist. Die Tasks beenden sich immer nur nach dem Ende eines Zyklusses, und nie mittendrin. n kann nicht größer als der Maximalwert (DMAXT=) werden, und auch nicht kleiner als 1. Wird einer dieser Werte angegeben, so wird ohne Fehlermeldung max. bzw. 1 angenommen.

Wenn der 1. Task ein Verwaltungstask ist, dann kann n nicht kleiner als 2 werden.

xxxxxxx     Datenbeschreibung

Quittungen am Terminal:

CISDBH - O.K.	Kommando ausgeführt.
CISDBH - ERROR	Fehler bei /ENTER.

**5.6.14 DOP Datei(en) wieder eröffnen**

```
$D DOP ,DB .xxxxxxx
```

\$D Operaton für "Diagnose".

DOP Operationsergänzung für Datei(en) eröffnen (OPEN). Dieses Kommando geht an einen der CISDBH - Tasks (bzw. an den einzigen CISDBH - Task). Die HD (und die VD) in diesem CISDBH - Task, die wieder eröffnet werden sollen, müssen vorher mit dem \$D DCL - Kommando geschlossen worden sein. Anschließend werden die anderen CISDBH - Tasks aufgerufen, die Datei(en) zu öffnen.

Jeder Task gibt eine positive oder negative Rückmeldung. Eine Rückmeldung mit dem Code FI26 ist kein Fehler, da dieser CISDBH - Task die Datei noch nicht gekannt hat (und sie somit auch beim \$D DCL nicht geschlossen wurde). Es wird jedoch nur sehr kurz auf Antworten gewartet. Sie können jederzeit mit \$D DCM abgefragt werden.

DB .xxxxxxx Das Datenbankpaßwort muß angegeben werden, um die HD und die eventuell vorhandene VD bekannt zu machen.

Quittungen am Terminal:

CISDBH - TSN: tttt - O.K.	Alles O.K. im Task tttt.
CISDBH - TSN: tttt - NO CM	Noch keine Antwort.
CISDBH - TSN: tttt - ERROR: eeee	Fehler eeee im Task tttt.

**5.6.15 DSEC Datei(en) vorreservieren**

```
$D DSEC ,DB .xxxxxxx
```

\$D Operation für "Diagnose".

DSEC Operationsergänzung für Datei(en) vorreservieren (SECURE). Dieses Kommando geht nur an einen der CISDBH - Tasks (bzw. an den einzigen CISDBH - Task). Die CISKOR - Sperrverwaltung wird aufgefordert, die HD so zu reservieren, daß keine neue Transaktion etwas in dieser Datei sperren kann. Diese Funktion kann benützt werden, wenn das Kommando \$D DCL,DB.xxxxxx die Fehlermeldung SP04 gebracht hat.

DB .xxxxxxx Das Datenbankpaßwort muß angegeben werden, um die HD und die eventuell vorhandene VD bekannt zu machen.

Quittungen am Terminal:

CISDBH - O.K.	Alles O.K.
CISDBH - TSN: tttt - ERROR: eeee	Fehler eeee im Task tttt.

### 5.6.16 DSTAT    Ausgabe der Statistik ein- und ausschalten

```
$D DSTAT J/Y/N[ ,DB .xxxxxxx ]
```

\$D            Operation für "Diagnose".

DSTAT        Operationsergänzung für Statistik ein- /ausschalten. Das Format der Statistikausgabe wird nicht erläutert. Es sind Diagnoseinformationen für die CIS - Beratung.

J/Y          Statistik soll bei Programmende ausgegeben werden.

N            Statistik soll nicht ausgegeben werden (Standardfall).

xxxxxxx     Datenbeschreibung

Quittungen am Terminal:

CISDBH - O.K.	Kommando ausgeführt.
CISDBH - ERROR	Falscher Wert eingegeben.

### 5.6.17 DTERMD    Dump und Ende von allen DBH - Tasks

```
$D DTERMD[ ,DB .xxxxxxx ]
```

\$D            Operation für "Diagnose".

DTERMD        Operationsergänzung für Dump und Ende. Es wird in allen Tasks dieses DBHs ein Dump ausgegeben. Anschließend werden die Programme beendet. Die Dumps müssen mit SODUMP aufbereitet werden.

Das Kommando wird in einem der DBH - Tasks bearbeitet. Der Dump als 2. Überschrift:

DIAGNOSTIC DUMP: CISDBH1

Im SYSOUT - Protokoll erscheint der Code CISDBH1. Die anderen DBH - Tasks erhalten ein Signal, das eine Contingency - Routine veranlaßt zu dumpen mit dem Code CISDBH2 und der 2. Überschrift:

DIAGNOSTIC DUMP: CISDBH2.

xxxxxxx     Datenbeschreibung

Quittung am Terminal:

CISDBH - O.K.	Kommando ausgeführt.
---------------	----------------------

### 5.6.18 DUNSEC Vorreservierung auflösen

```
$D DUNSEC [ ,DB .xxxxxxx ]
```

\$D Operation für "Diagnose".

DUNSEC Operationsergänzung für Vorreservierung auflösen. Dieses Kommando geht nur an einen der CISDBH - Tasks (bzw. an den einzigen CISDBH - Task). Die CISKOOR - Sperrverwaltung wird aufgefordert, die von dieser Transaktion vorreservierten HDs wieder freizugeben. \$D DUNSEC muß also vom gleichen Terminal gegeben werden wie \$D DSEC,DB.xxxxxx (bzw. beide mit /INTR).

xxxxxxx Datenbeschreibung

Quittungen am Terminal:

CISDBH - O.K.	Alles O.K.
CISDBH - TSN: tttt - ERROR: eeee	Fehler eeee im Task tttt.

### 5.6.19 DWAIT Warten

```
$D DWAIT n [ ,DB .xxxxxxx ]
```

\$D Operation für "Diagnose".

DWAIT Operationsergänzung für Warten

n Anzahl Sekunden, die zu warten ist.

xxxxxxx Datenbeschreibung

Quittungen am Terminal:

CISDBH - O.K.	Kommando ausgeführt	.
---------------	---------------------	---

## 5.7 Kommandos für CISKURZ

### 5.7.1 Übersicht

Kommando	Erläuterung
<code>;komm - 1;komm - 2; ... ;komm - n</code>	Kommandokettung
<code>%CISGEN kommando</code>	CISGEN - Aufruf
<code>%CTGET,F[B],feldbezeichnung</code>	CIS - Passivkommando
<code>%GET BUFFER+d typ länge</code>	Unbenannte Variable aus EB setzen.
<code>%MODIFY BUFFER+d=...</code>	EB setzen.
<code>%MODIFY LENGTH=...</code>	LEB setzen.
<code>%PASSIV OFF</code>	Passiv - Betrieb ausschalten.
<code>%PASSIV ON</code>	Passiv - Betrieb einschalten.
<code>%SEND EVENTING text (l=1-22)</code>	POSSIG mit Item + text
<code>%SEND SETUS ON=nn                   OF=nn</code>	Setzen Userschalter. Rücksetzen Userschalter.
<code>%SHOW BUFFER+d...</code>	EB ausgeben.
<code>%SHOW LENGTH</code>	LEB ausgeben.
<code>%SYS [/BS2 - Kommando]</code>	Breakpoint bzw. BS2 - Kommando.
<code>%WAIT n</code>	VPASS n
<code>%WAIT EVENTALL text (l=1-22)</code>	SOLSIG mit Item + text (ohne Zeitlimit)
<code>%WAIT EVENTALL text (l=1-22)</code>	SOLSIG mit Item + text (max. 6 Minuten)
<code>%WAIT GETUS nn</code>	Warten auf Userschalter nn
<code>%WHEN EVENTING text:key1#key2</code>	Prüfen Event - Item text und springen.
<code>%WHEN GETUS nn:key1#key2</code>	Prüfen Userschalter nn und springen.

### 5.7.2 Kommandokettung

```
;komm - 1;komm - 2;...
```

Es können mehrere CIS - Kommandos und Kurzkommandos eingegeben werden. Diese werden so gespeichert als ob sie aus einer Kurzkommandoprozedur gelesen worden wären. Sie sind ab 1 mit einer Schrittweite von 1 durchnummeriert. Es wird das erste Kommando gestartet.

### 5.7.3 %CISGEN

#### CISGEN Aufruf

```
%CISGEN kommando
```

%CISGEN      Operation für "CISGEN - Aufruf".

kommando    CISGEN - Kommando.

**5.7.4 %CTGET**

**CIS Passivkommando**

```
%CTGET , F [ B ] , feldbezeichnung
```

%CT            Operation für "Bearbeiten Passiv Kommando"

GET , FB       Die Feldbedeutung des angegebenen Feldes soll ausgegeben werden.

GET , F        Ein Feldinhalt des angegebenen Feldes (auch mit evtl. angegebenen Ergänzungen (AM/FM)) soll ausgegeben werden.

feldbezeichnung

Feld auf das sich die Fragestellung bezieht.

Aus LZI und ZI wird im Übergabebereich ein V - Satz aufgebaut:

```
00L3 4040 000000L2 00L1 4040 ZI.....  
  1 2  3 4  5 6 7 8  1 2  3 4  . . . . .
```

L1            LZI + 4 für V - Satzlänge(2) und Space(2)

L2            LZI + 4 Satzlänge der Ausgabe (Zeige - Standard)

L3            LZI + 12 Gesamtlänge mit 8 Byte Vorspann

ZI            Wert aus ZI, den CIS geliefert hat.

Das Kommando hat im normalen KUKO - Ablauf keinen Sinn, da mit dem ausgegebenen Format kein äußeres Modul (CISKURZ1) etwas anfangen kann.

Fehlermeldungen werden sofort und unter Umgehung der üblichen CISKURZ - Logik zum Ausgang durchgereicht.

### 5.7.5 %GET BUFFER Unbenannte Variable aus EB setzen

```
%GET BUFFER[+d] B1
                  C1
                  H1[,dez]
                  P1[,dez]
                  R1[,dez]
```

**%GET BUFFER** Operation für "@ aus EB setzen"

**d** Distanz in EB (dezimal oder hex: X'd'). Wenn Angabe fehlt, dann wird 0 angenommen.

**B/C/H/P/R** Format des Feldes: - B: Binär  
 - C: Zeichen (character)  
 - H: Gepackt ohne Vorzeichen  
 - P: Gepackt  
 - R: Rechtsbündig

**l** Länge des Feldes in Bytes

**dez** Anzahl Dezimalstellen (Formate H/P/R)

### 5.7.6 %MODIFY BUFFER EB setzen

```
%MODIFY BUFFER[+d]=B[l]'wert'
                  C
                  H
                  P
                  R
                  X
```

**%MODIFY BUFFER** Operation für "EB setzen"

**d** Distanz in EB (dezimal oder hex: X'd') wenn Angabe fehlt, dann wird 0 angenommen.

**B/C/H/P/R/X** Format des Feldes: - B: binär  
 - C: Zeichen (character)  
 - H: gepackt ohne Vorzeichen  
 - P: gepackt  
 - R: rechtsbündig  
 - X: hexa

**l** Länge des Feldes in Bytes. Es wird formatgerecht eingerichtet, nicht als Beschränkung.

z.B. C2'AAAAA' überschreibt 5 Bytes mit A  
 C5'AA' löscht 5 Bytes mit Space trägt dann 2 Bytes AA ein.

**wert** Neuer Wert (externes Format)

### 5.7.7 %MODIFY LENGTH LEB setzen

```
%MODIFY LENGTH=l  
X'hhh'
```

%MODIFY LENGTH Operation für "LEB setzen".

l Dezimaler Wert für LEB.

hhh Hexa - Wert für LEB, immer 4 Zeichen.

LEB enthält nach Init 32KB, sonst immer den Wert des letzten GET - Befehls.

### 5.7.8 %PASSIV OFF Passivbetrieb ausschalten

```
%PASSIV OFF
```

%PASSIV Operation für "Passiv - Betrieb"

OFF Ausschalten

### 5.7.9 %PASSIV ON Passivbetrieb einschalten

```
%PASSIV ON
```

%PASSIV Operation für "Passiv - Betrieb".

ON Einschalten

### 5.7.10 %SEND

### Senden / Setzen bestimmter Ereignisse

```
%SEND SETUS ON=nn
        OFF=nn
        EVENTING text (l=1 - 22)
```

%SEND	Operation für "Senden".
SETUS ON=nn OFF=nn	Der Userschalter mit der Nummer nn wird gesetzt. Der Userschalter mit der Nummer nn wird rückgesetzt.  Die Nummer kann ein- oder zweistellig angegeben werden. Eine Fehlerabfrage, ob der BS2000 SETUS erfolgreich gelaufen ist, erfolgt nicht.
EVENTING	Das BS2000 Ereignis CISKURZ@WAITEVENTING@EVENTING@text wird erzeugt.  Dazu wird ein ENAEI und mindestens ein POSSIG und ein CHKEI abgesetzt.  Es werden so lange POSSIG's abgegeben, bis der nachfolgende CHKEI eine leere Warteschlange signalisiert.  ENAEI verzweigt im Fehlerfall auf KU50 (abfragbar). Der Fehlercode aus Register 15 kommt als Ergänzung.  POSSIG verzweigt im Fehlerfall auf KU51 (abfragbar). Der Fehlercode aus Register 15 kommt als Ergänzung.  CHKEI verzweigt im Fehlerfall auf KU55 (abfragbar). Der Fehlercode aus Register 15 kommt als Ergänzung.  Das Ereignis ist global. Die Wartezeit ist 600 Sekunden.  Alle Syntaxfehler im %SEND führen zum Abbruch mit KU52.

### 5.7.11 %SHOW BUFFER EB ausgeben

```
%SHOW BUFFER[+d]C/X[l]
```

%SHOW BUFFER	Operation für "EB ausgeben".
d	Distanz in EB (dezimal oder hex: X'd'). Wenn die Angabe fehlt, dann wird 0 angenommen.
C/X	Format des Feldes: - C: Zeichen (character) - X: hexa
l	Länge des Feldes in Bytes.

### 5.7.12 %SHOW LENGTH EB ausgeben

```
%SHOW LENGTH
```

%SHOW LENGTH Operation für "LEB ausgeben".

### 5.7.13 SYS BS2000 System Kommando

```
%SYS [/BS2 - Kommando]
```

%SYS Operation für Systemmodus.

/BS2 - Kommando BS2000 - Kommando, das ausgeführt werden soll. Das Kommando kann maximal 200 Zeichen lang sein. AID und IDA Kommandos werden zurückgewiesen. \$ statt / schaltet diese Prüfung aus. Fehlt ein BS2000 - Kommando so wird im TIAM mit BKPT in die Kommandoebene BS2000 verzweigt. Bei Syntaxfehlern und im Falle der Zurückweisung wird KU46 gesetzt und eine evtl. Kommandoprozedur unterbrochen. Der Fehler bei Kommandoausführung ist KU47 und ist in der Prozedur mit %CM abfragbar.

### 5.7.14 %WAIT

### Warten auf bestimmte Ereignisse

```
%WAIT nnn
      GETUS nn
      EVENTING text (l=1-22)
      EVENTALL text (l=1-22)
```

%WAIT	Operation für "Warten".
nnn	1 - 4 stellige Ziffernangabe als Sekunden - Wert für den VPASS.
GETUS nn	Es soll gewartet werden bis der Userschalter mit der Nummer nn gesetzt ist. Die Nummer nn kann ein oder zweistellig angegeben werden. In der GETUS Schleife wird ein VPASS 0 abgesetzt und ein Zähler für die Fehlversuche erhöht. Nach 1024 Versuchen wird mit KU53 (abfragbar) abgebrochen.
EVENTING	<p>Es soll auf das Ereignis CISKURZ@WAITEVENTING@EVENTING@text gewartet werden.</p> <p>Dazu wird ein ENAEI und ein SOLSIG abgesetzt.</p> <p>ENAEI verzweigt im Fehlerfall auf KU48 (abfragbar). Der Fehlercode aus Register 15 kommt als Ergänzung.</p> <p>SOLSIG verzweigt im Fehlerfall auf KU49 (abfragbar). Der Fehlercode aus Register 15 kommt als Ergänzung.</p> <p>Das Ereignis ist global. Die Wartezeit ist 600 Sekunden.</p> <p>Alle Syntaxfehler im %WAIT führen zum Abbruch mit KU54.</p>
EVENTALL	Wie EVENTING, jedoch ist die Wartezeit unendlich.

### 5.7.15 %WHEN                    Prüfen bestimmter Ereignisse

```
%WHEN GETUS nn:key1#key2  
      EVENTING text:key1#key2
```

%WHEN                    Operation für "Prüfen"

GETUS nn                Es wird geprüft ob der Userschalter mit der Nummer nn gesetzt ist. Die Nummer nn kann ein oder zweistellig angegeben werden. Ist der Userschalter gesetzt, so wird bei key1 fortgesetzt, im anderen Fall wird nach key2 verzweigt.

EVENTING                Es wird geprüft, ob mindestens ein Possig für das Ereignis CISKURZ@WAITEVENTING@EVENTING@text vorhanden ist. Wenn ja, so wird bei key1 fortgesetzt, im anderen Fall wird nach key2 verzweigt.

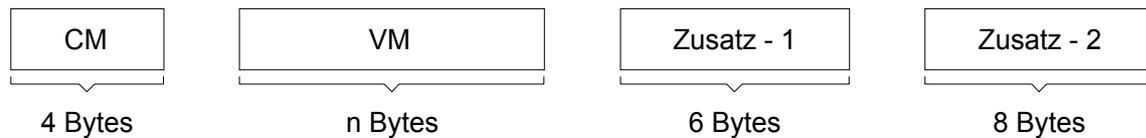
ENAEI verzweigt im Fehlerfall auf KU57 (abfragbar). Der Fehlercode aus Register 15 kommt als Ergaenzung.

Alle Syntaxfehler im %WHEN führen zum Abbruch mit KU56.

## 6 Erläuterungen zu den CO - Fehlermeldungen

### 6.1 Format der Fehlermeldungen

Die CO - Fehlermeldung (CM - Format: COnn) betreffen das Kommunikationsmodul COMM oder CISKOOR.



CM = Codierte Meldung, Code COnn  
 VM = Verbale Meldung, Text

Zusatz - 1: 6 Bytes langer abdruckbarer Zusatz.

Zusatz - 2: 8 Bytes langer sedezimaler Zusatz (aufbereitet, also Inhalt von 4 Bytes).

### 6.2 Übersicht über die Fehlermeldungen

Die Fehlermeldungen mit den dazugehörenden Erläuterungen können mit dem CISGEN - Kommando LSTE ausgedruckt werden.

### 6.3 Übersicht der Rückmeldungen von CISKOOR

Zusatz - 2		Bedeutung
4 linke Zeichen	4 rechte Zeichen	
0003 0003 0003	00aa Dxxx FFFF	AIM - Fehler: Werte von aa: Vgl. Fehlermeldungen CISAIM - Seite 161. AIM - DMS Fehler (Systemmeldung Dxxxx). AIM in CISKOOR nicht generiert.
0004 0004 0004	00bb Dxxx FFFF	BIM - Fehler: Werte von bb: Vgl. Fehlermeldungen CISBIM - Seite 162. BIM - DMS Fehler (Systemmeldung Dxxx). BIM in CISKOOR nicht generiert.
0005 0005 0005	00pp Dxxx FFFF	PRO - Fehler: Werte von pp: Vgl. Fehlermeldungen CISPRO - Seite 163. PRO - DMS Fehler (Systemmeldung Dxxx). PRO in CISKOOR nicht generiert.
0006 0006 0006  0006 0006	00bb Dxxx F3F2  F3F3 F3F4	Restart BIM Fehler: Werte von bb: Vgl. Fehlermeldungen CISBIM - Seite 162. BIM - DMS Fehler beim Restart. (Systemmeldung Dxxx). Tabelle voll: Wenn mit Transaktionen gearbeitet wird dann ist die Transaktions- tabelle voll. Sie hat 200 Einträge.  Wenn ohne Transaktionen gearbeitet wird, dann ist die TSN - Tabelle voll. Sie hat 100 Einträge.  Die Transaktion ist CISKOOR nicht bekannt. "Transaktionen" in CISKOOR nicht generiert.
0007  0007	ssee  FFFF	Fehler beim Sperren: Werte von ss Vgl. Fehlermeldungen CISSP - Seite 163. Werte von ee x'00' oder EAM - Fehler Byte. Vgl. "BS2000 Datenverwaltungssystem, Abschnitt über EAM"  "Sperren" nicht generiert.
0008  0008	ssee  FFFF	Fehler beim Entsperrn: Werte von ss Vgl. Fehlermeldungen CISSP - Seite 163. Werte von ee x'00' oder EAM - Fehler Byte. Vgl. "BS2000 Datenverwaltungssystem, Abschnitt über EAM"  "Sperren" nicht generiert.
000A  000A	ssee  FFFF	Fehler beim Sperrenabfragen: Werte von ss Vgl. Fehlermeldungen CISSP - Seite 163. Werte von ee x'00' oder EAM - Fehler Byte. Vgl. "BS2000 Datenverwaltungssystem, Abschnitt über EAM"  "Sperren" nicht generiert.

## 6.4 Fehlermeldungen von CISAIM

X'aa'	Bedeutung
X'01'	Kennung falsch.
X'02'	OPEN - OUTPUT, aber Datei nicht geschlossen.
X'03'	OPEN - INPUT, aber Datei nicht geschlossen.
X'05'	CLOSE, aber Datei nicht eröffnet.
X'06'	Schreiben, aber datei nicht OUTPUT - eröffnet.
X'07'	Lesen, aber kein OPEN - INPUT.
X'08'	UPDATE, aber kein OPEN - INPUT.
X'09'	Initialisieren, aber Datei nicht geschlossen.
X'0A'	Keine After - Image Datei.
X'0B'	Beim Lesen Satz zu groß.
X'0C'	Fehler bei REQM 1. PAGE - Bereich.
X'0E'	Fehler bei REQM 1. PAGE - Bereich.
X'10'	Falsche KOOR - ID.
X'11'	Alte After - Image Datei.

**6.5 Fehlermeldungen von CISBIM**

X'bb'	Bedeutung
X'41'	Kennung falsch.
X'49'	Falsche KOOR - ID
X'4A'	Keine Before - Image Datei.
X'4B'	RESTART muß gefahren werden.
X'4D'	Kein Platz für Tabellen.
X'4E'	Kein Platz mehr in der Dateitabelle.
X'51'	Fehler beim Freigeben Tabellen.
X'52'	Fehler beim Freigeben BUFFER.
X'53'	Kein Platz für BUFFER.
X'55'	Kein freier FCB mehr.
X'57'	Fehler bei REQM PAM Bereiche.
X'59'	Fehler bei RELM PAM Bereiche.
X'5A'	ENASI - Fehler. SE01
X'5B'	ENQAR - Fehler. SE02
X'5C'	DEQAR - Fehler. SE03
X'5D'	DISSI - Fehler. SE04
X'5E'	Falsche Kennung. SE05
X'5F'	Zu langes Warten auf Sperren. SE06
X'60'	US - Nr. zu groß in VD. UP08
X'61'	VD - FCB fehlt. UP18
X'62'	OPCODE - ERROR in UVD. UP19
X'63'	VD zerstört. UP22
X'64'	Sonstiger Fehler in UVD.
X'65'	Statustabelle voll.
X'66'	Statussatz voll.
X'68'	CLOSE : noch BIMs vorhanden.
X'6A'	INIT : PRIM. ALLOC. nicht $\geq 2$ * BIM - Parameter.
X'6B'	INIT : SEC. ALLOC. nicht $\geq 2$ * BIM - Parameter.
X'6C'	OPEN : Alte BIM - Datei leer.
X'6D'	OPEN : Alte BIM - Datei nicht leer.

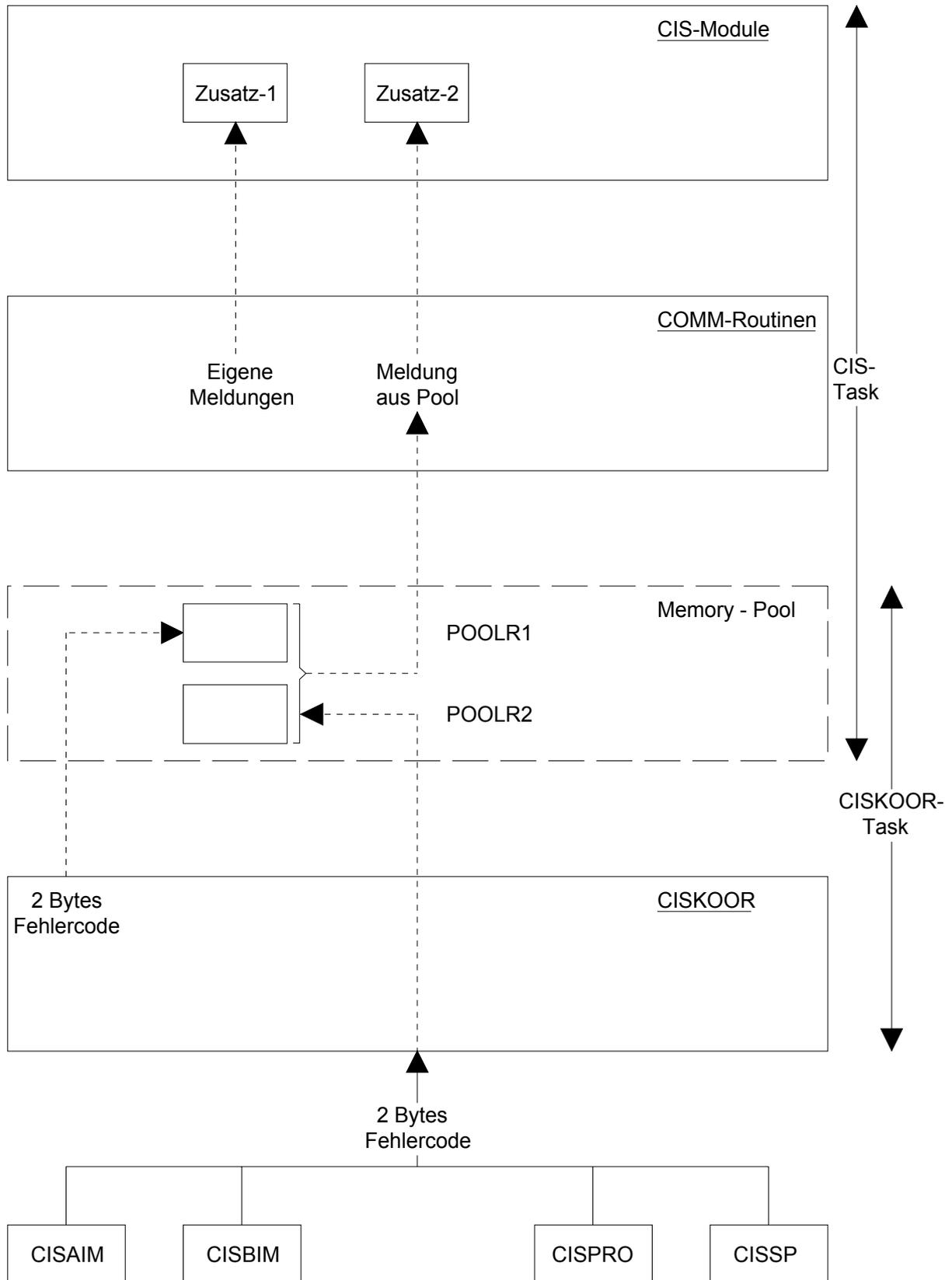
## 6.6 Fehlermeldungen von CISPRO

X'pp'	BEDEUTUNG
X'81'	Kennung falsch.
X'89'	Falsche KOOR - ID.
X'8A'	Keine Protokolldatei (Inhalt 1. Satz).
X'8B'	Beim Lesen Satz zu groß.
X'8C'	Keine Protokolldatei (Nicht ISAM - Datei).

## 6.7 Fehlermeldungen von CISSP

X'ss'	Bedeutung.
X'C1'	Falsche Kennung.
X'C2'	Kein Platz in der Tabelle.
X'C5'	EAM - Fehler: OPEN.
X'C6'	EAM - Fehler: Schreiben.
X'C7'	EAM - Fehler: Lesen.
X'C8'	EAM - Fehler: Erase - Datei.
X'CC'	Nichts gesperrt. (Entsp.)
X'CD'	Datei als Ganzes gesperrt. (Entsp. Liste)
X'CE'	Sätze in der Datei gesperrt. (Entsp. Datei)
X'CF'	Fehler beim REQM. (ENABLE)
X'D0'	Fehler beim RELM. (DISABLE)
X'D1'	Nicht gesperrt. (Prüfen)
X'D4'	Fehler bei FSTAT.

### 6.8 Hierarchie des Fehlermeldewesens



## 7 Diverse Beschreibungen zu CIS

### 7.1 Beschreibung der VMADL

(siehe auch Makros VMADD und VMEXT)

Dies ist die zentrale Adressliste von CIS. Sie ist in IVMD definiert.

Bytes		Länge	Inhalt
dezimal	hex.		
0 - 3	0 - 3	4	Adresse der Kommandolänge.
4 - 7	4 - 7	4	Adresse des Kommandos (Kommando des Aufrufers - verbal oder formatiert).
8 - 11	8 - B	4	Adresse des Feldes für die Codierte Meldung.
12 - 15	C - F	4	Adresse des Feldes für die verbale Meldung.
16 - 19	10 - 13	4	Adresse der Länge der Zielinformation.
20 - 23	14 - 17	4	Adresse des Feldes für die Zielinformation.
24 - 27	18 - 1B	4	FCBGET
28 - 31	1C - 1F	4	Adresse von GETBER (Einlesebereich).
32 - 35	20 - 23	4	Adresse des formatierten Auftrags.
36 - 39	24 - 27	4	ADB (Adresse Abschnitt).
40 - 43	28 - 2B	4	Adresse ZPL - Kopf.
44 - 47	2C - 2F	4	Adresse VD - FCB.
48 - 51	30 - 33	4	Adresse HD - FCB.
52 - 55	34 - 37	4	Adresse private Schalter
56 - 59	38 - 3B	4	Adresse des Kommunikationsfeldes KOMFELD in IVMD.
60 - 63	3C - 3F	4	Adresse aktuelle Zielpunktliste (auf ZPLIST1 vorbesetzt). Die andere Zielpunktliste kann z. B. als Ein/Aus - Bereich von 2K benützt werden.
64 - 67	40 - 43	4	Adresse 1. Zielpunktlistenbereich.
68 - 71	44 - 47	4	Adresse 2. Zielpunktlistenbereich.
72 - 75	48 - 4B	4	Adresse Arbeitsspeicher.
76 - 79	4C - 4F	4	Adresse Länge ASPEICH.
80 - 83	50 - 53	4	Adresse Länge IO - Bereich.
84 - 87	54 - 57	4	Adresse Länge formatierter Bereich.
88 - 91	58 - 5B	4	Adresse DBLADL in IVMD. DBLADL: 4 Bytes Adresse DBNAME (6 Bytes). 4 Bytes Adresse Fehladresse DBL (4 Bytes).
92 - 95	5C - 5F	4	READB
96 - 99	60 - 63	4	REZI

Ab hier im Makro VMEXT beschrieben.

## 7.2 Formatierter Auftrag

### Kommandolängenfeld

Bytes dezimal   hex.		Länge	Format	Inhalt
0 - 1	0 - 1	2	B	Länge aller Abschnitte + 4
2 - 3	2 - 3	2	x	Kettungsteil: x'4040' = geschlossene Kette x'FFFF' = offene Kette x'0000' = Kettenende

### Kopfabschnitt

Bytes dezimal   hex.		Länge	Format	Inhalt
0 - 1	0 - 1	2	B	Abschnittslänge (22=x'16')
2 - 3	2 - 3	2		Kettung (?)
4 - 7	4 - 7	4	C	Abschnittsname KOPF
8 - 9	8 - 9	2	X	Modulkennung: 1. Byte: Modulnummer 2. Byte Operation
10 - 11	A - B	2	X	Unterprogrammkenung: 1. Byte: UP - Nr. 2. Byte: Typ
12 - 15	C - F	4	B	Menge (\$ = 999999 = x'F423F')
16 - 21	10 - 15	6	C	Paßwort

## Feldabschnitt

Bytes		Länge	Format	Inhalt
dezimal	hex.			
0 - 1	0 - 1	2	B	Abschnittslänge
2 - 2	2 - 2	1	B	Low - Value oder Joinnummer bei 78 er - Auftrag.
3 - 3	3 - 3	1	B	Länge Feldbezeichnung.
4 - 7	4 - 7	4	C	ASPE (Abschnittsname).
8 - 11	8 - B	4	X	XL4'00'(Absolute Adresse D - Segment).
12	B + <i>l</i>	<i>l</i>	C	Feldbezeichnung

## Ergänzungsabschnitt

Bytes		Länge	Format	Inhalt
dezimal	hex.			
0 - 1	0 - 1	2	B	Abschnittslänge
2 - 3	2 - 3	2		Reserve XL2'00'
4 - 7	4 - 7	4	C	ERGE (Abschnittsname)
8 - 9	8 - 9	2	B	Feldlänge (FL - Param.) +
10 - 11	A - B	2	B	Feldbedeutung (FB - Param.)
12 - 13	C - D	2	B	Feldposition (FP - Param.) + -
14 - 15	E - F	2		Reserve XL2'00'
16 - 19	10 - 13	4	B	Abschnittsmultiplizität (AM - Param.) + -
20 - 23	14 - 17	4	B	Feldmultiplizität (FM - Param.) + -
24 - 27	18 - 1B	4	B	Abschnittszyklen (AZ - Param.) +
28 - 31	1C - 1F	4	B	Feldzyklen (FZ - Param.) +
32 - 32	20 - 20	1	B	Sortierkennzeichen: S,F,US,UF = 0,4,8,C
33 - 38	21 - 26	6	C	Paßwort wenn (DB=...)

### 7.3 Beschreibung des Kommunikationsfeldes

(siehe auch Makro KOMFLD)

- Das Feld
- Ist in VIVMD definiert.
  - Ist 11 Worte groß (44 Bytes).
  - Wird adressiert über VMADL ( + x'38' ).

Bytes		Länge	Inhalt
dezimal	hex.		
0 - 3	0 - 3	4	Adresse DB - Name: Verweist auf ein Feld von 6 Bytes mit dem DB - Namen
4 - 4	4 - 4	1	Kommandowechselprüfung : Wird in IVMD gesetzt / rückgesetzt. x'00': Kommandowechsel. x'FF': Kein Kommandowechsel.
5 - 7	5 - 7	3	Reserve
8 - 11	8 - B	4	Kettenkennungen: Byte 8: "PUT,K" - offen gesetzt in IUPD auf C'P' wenn PUT,K - offen rückgesetzt in IUPD auf x'00' am Ende und in IVMD bei IGN,K.  Byte 9: "Kette bereits offen" gesetzt in IUPD auf x'01' am Anfang offene Kette rückgesetzt in IUPD auf x'00' am Ende und in IVMD bei IGN,K.  Bytes 10 - 11: Reserve
12 - 15	C - F	4	Diverse Kennungen Byte 12: Sperr - Bits (Kommando SE,SP) x'80' Sperre gesetzt x'40' Sperre aktiv x'20' "Echter" IGET - Aufruf. x'10' Aufruf IGET von IUPD  Byte 13: Sperr - Bits (Kommando SP,Z) x'80' Etwas gesperrt.  Byte 14: Transaktions - Bits (Kommando TR) x'80' Es war TR,A  Byte 15: Bits für ISUCH x'80' Es war SE,S x'40' Es war SI,S
16 - 21	10 - 15	6	Vorhergehendes Paßwort in ISUCH.
22 - 23	16 - 17	2	Reserve (Ausrichtung auf nächstes Wort.)
24 - 27	18 - 1B	4	Adresse vorhergehende Datenbeschreibung (SYNTAX).
28 - 31	1C - 1F	4	Diverse Kennungen - 2 Byte 28: ZNSWITCH in SYNTAX. Byte 29: Erlaubt - Kennzeichen in IDIAG Byte 30: Reserve Byte 31: Reserve
32 - 35	20 - 23	4	IAUS - Zähler
36 - 39	24 - 27	4	Adresse vorhergehende Datenbeschreibung (IUPD).
40 - 43	28 - 2B	4	Satzart, zu der aktuelle ZPL paßt (IGET, GETD).

## 7.4 Beschreibung des Zielpunktkopfes

Das Feld ZPLKOPF: - Ist in IVMD definiert.  
 - Ist auf Wortgrenze ausgerichtet.  
 - Ist 48 Byte groß.  
 - Wird adressiert über VMADL (Adr. in VMADL + x'28').

Siehe auch Makro ZPLKOPF

Bytes		Länge	Inhalt
dezimal	hex.		
0 - 1	0 - 1	2	ZPL - Typ H'0' H'8' ZPL - Kopf - Kopie im ZPL - Keller, EAM - Datei geschützt
2 - 2	2 - 2	1	Sperr - Bits: x'80' ZPL gesperrt. x'40' ZPL sortiert.
3 - 3	3 - 3	1	Status: x'00' ZPL steht im Speicher. x'02' ZPL mehrspaltig (JOIN - ZPL). x'04' ZPL steht im EAM - Bereich. x'08' ZPL ist in VD gespeichert. x'10' ZPL mit Verdichtung. x'20' ZPL steht auf verdichtetem Eintrag. x'FF' ZPL wurde ignoriert.
4 - 7	4 - 7	4	Anzahl Zielinformationen.
8 - 11	8 - B	4	AZI aktuell (Pointer in AZI).
12 - 13	C - D	2	Länge der Zieladresse (= Länge der HD - ISAM - Keys.)
14 - 15	E - F	2	Länge des Zieladressenzusatzes (=Länge des Sortiebegriffs). Nur nach Sortieren der Zielpunktliste. Format des Eintrags: - Zusatz (Sortierbegriff) - Zieladresse
16 - 19	10 - 13	4	Nummer aktuelle Zieladresse.
20 - 23	14 - 17	4	frei
24 - 25	18 - 19	2	Untersatznummer: Beim Lesen der ZPL aus EAM über Modul ZPV muß hier die zu lesende Untersatznummer eingetragen werden.
26 - 27	1A - 1B	2	Adresse innerhalb des ZPL - Satzes.
28 - 31	1C - 1F	4	Adresse der ZPL wenn diese im Speicher steht (wenn Status = 0).
32 - 33	20 - 21	2	EAM - Blocknummer
34 - 35	22 - 23	2	Nummer der EAM - Datei wenn Status = 4
36 - 37	24 - 25	2	Anzahl EAM - Blöcke wenn Status = 4
38 - 39	26 - 27	2	Länge des ZPL - Keys (= Länge des ISAM - Keys in der VD).
40 - 43	28 - 2B	4	Adresse des FCB's der betroffenen HD (Die HD, in der die ISAM - Keys gesucht werden.).
44 - 47	2C - 2F	4	Binärer Key (Ein- bis vierstellig) wenn Verdichtung läuft.

## 7.5 Beschreibung des Zielpunktlistenkellers

Bereich: TIAM: in VIVMD ZPKELLER  
 UTM: im Terminal - Bereich  
 Anfangsadresse: In der VMADL AZPKELL  
 Länge: In der VMADL LZPKELL = 2000

Jeder Eintrag hat folgendes Format:

Bytes			
von	bis		
0	1	2	Länge des Paßwortes (6) plus Länge der Feldbezeichnung (a).
2	7+a	6+a	Paßwort (6 Bytes) Feldbezeichnung (a Bytes) wie in der Datenbeschreibung angegeben. @@@@ bei SI,S @@@f wenn Fenster
8+a	9+a	2	Länge des Feldinhalts (w).
10+a	9+a+w	w	Feldinhalt: Länge und Format wie in der Datenbeschreibung angegeben. Wenn Fenster, dann mehrere Join - Spalten - Bereiche (siehe DSECT IJAUFTR).
10+a+w	11+a+w	2	Länge ZPL - Kopf + Ergänzungen.
12+a+w	11+a+w+l	l	ZPL - Kopf + Ergänzungen.

Beispiel: Es gibt keine Ergänzungen (keine Y - Rechenregister) wenn mit einem Feld mit der Bezeichnung EDV - NR, das als 3 - stellige Binärzahl definiert ist, gearbeitet wird.

Es ergibt sich:  $\left. \begin{matrix} a = 6 \\ w = 3 \end{matrix} \right\} \longrightarrow \blacktriangleright$  Eintragslänge = 69 Bytes

In den Keller passen somit:  $2000 : 69 = 28$  Einträge

## 7.6 Beschreibung eines Extraktionsatzes

Bytes		Länge	Format	Inhalt	
von	bis				
0	1	2	B	Länge des Satzes	
2	2	1		Filler	
3	3	1	X	Operationscode *)	x'00' Einfügen - IUPD x'01' Löschen - IUPD . . x'10' Verdichten möglich x'20' Neuer EX - Satz.
4	6	3	C	Segment (Kurzname (Kurzaspekt))	bis V10
4	7	4	C	Segment (Kurzname) (wenn OP = X'20')	ab V11
7	7	1	B	Länge Ordnungsbegriff	bis V10
8	8	1	B		ab V11
8	7+x	x		Ordnungsbegriff (= ISAM - Schlüssel der HD)	bis V10
9	8+x	x			ab V11
8+x	8+x	1	B	Länge des Feldinhalts des invertierten Feldes, der in den	bis V10
9+x	9+x	1	B	ISAM - Schlüssel der VD eingeht. (Deskriptor)	ab V11
8+x+1	8+x+y	y		Feldinhalt des invertierten Feldes, der in den ISAM -	ab V10
9+x+1	9+x+y	y		Schlüssel der VD eingeht.	ab V11

\*) siehe Source von UVD

## 7.7 Beschreibung eines VD - Satzes

Bytes		Länge	Format	Inhalt	
von	bis				
0	2	3	C	VD - Segmentname	VD - Key
3	2+a	a		Feldinhalt (a = MAXDES)	
3+a	6+a	4	B	Unterschlüssel ab x'0000001'möglich.	
7+a	6+a+d	d		Daten (Ordnungsbegriff)	
7+a+d	6+a+d+4	4	B	Folgesatzkennung:	F'0' Kein weiterer Untersatz. F'1' Es folgen Untersätze. F'1' Im ersten Untersatz wenn Untersätze folgen. (I=höchste Untersatznummer)

## 7.8 Liste der Kommandos und Kennungen

Es folgt eine alphabetische Liste der CIS - Kommandos, mit den jeweiligen Modul- und Unter - rogrammkennungen.

Kommando	Modul - UP#	KL	DIV.	Erläuterungen
AE, K	1404 - 0C00	00	P	Satz (Kapitel)
KS	1404 - 0C0C	00	PI	Satz - Spezial (Leasy)
KD	1404 - 0C10	00	P	Satz - direkt
A	1404 - 1004	16	P	Abschnitt
F	1404 - 1408	01	AP	Feld
KF	1404 - 1408	09	AP	Satz feldweise
AK	0C04 - 0400	01	AP	EAM
J	7804 - 0400	01	AP	"Joined ZPL" (V9)
E	0C04 - 0400	01	AP	EAM
I	0C04 - 0410	01	AP	VD - Pointer
Z	0C04 - 0400	01	AP	"ZPL" (historisch)
BI	14			Siehe: SUCHEN
BL, D	2004 - 0008	09	AP	Direkt d.h. Hauptdatei
I	2004 - 000C	09	AP	Indirekt d.h. Verweisdatei
V	2004 - 0000	00	AP	Vorwärts
R	2004 - 0004	00	AP	Rückwärts
C	2000 - 0000	12	AP	Aktuelle HD/VD
A	2000 - 0004	00	AP O	All
S	2000 - 0404	00	API	Spezial
DR	04			Siehe: ZEIGEN
EI, K	2C00 - 0000	00	A	Satz (Kapitel)
A	2C00 - 0404	00	A	Abschnitt
EN	1808 - 1404	00	AP	Kein Typ bei Ende.
EX, L	2404 - 0000	02	AP	Liste
A	7404 - 0000	02	AP	ADILOS
TA	7404 - 0004	02	AP	ADILOS
F	0C08 - 0800	01	AP	EAM
E	0C08 - 0800	01	AP	EAM
I	0C08 - 0810	01	AP	VD - Pointer
S	0C08 - 1804	00	AP O	SI, S aus + Listen freigeben
V	0C08 - 0804	01	AP	VD

Kommando	Modul - UP#	KL	DIV.	Erläuterungen
G ,K	0800 - 0000	00	P	Satz (Kapitel)
KP	0800 - 0000	00	P	Satz
KPS	0800 - 0008	00	P	Satz + Sperre
N	0800 - 0014	00	P	Nächster
KN	0800 - 0014	00	P	Nächster Satz
KNS	0800 - 001C	00	P	Nächster Satz + Sperre
KR	0800 - 0034	00	P	Nächster Satz rückwärts
KRS	0800 - 003C	00	P	Nächster Satz rückwärts + Sperre
KD	0800 - 1400	01	P	Satz direkt
KDS	0800 - 1408	01	P	Satz direkt + Sperre
KS	0800 - 1408	01	P	Satz direkt + Sperre
KFP	0800 - 1800	09	A	Satz feldweise
KFPS	0800 - 1808	09	A	Satz feldweise + Sperre
KFN	0800 - 1814	09	A	Nächster Satz feldweise
KFNS	0800 - 181C	09	A	Nächster Satz feldweise + Sperre
KFR	0800 - 1834	09	A	Satz rückwärts feldweise
KFRS	0800 - 183C	09	A	Satz rückwärts feldweise + Sperre
KI	0800 - 1C04	09	P	Satz über Verweisdatei direkt
KIS	0800 - 1C0C	09	P	Satz über Verweisdatei direkt + Sperre
V	0800 - 0C0C	01	P	VD - Satz
NV	0800 - 0C14	00	P	Nächste VD
ZP	0800 - 1000	10	P	Zielpunktliste ab Pointerstand
A	0800 - 0400	06	P	Abschnitt
AS	0800 - 0408	06	P	Abschnitt aus gesperrten Satz
F	0800 - 0800	01	P	Feld
FS	0800 - 0808	01	P	Feld aus gesperrtem Satz
B	1C00 - 0000	12	P	Bildschirm
P	1C00 - 0404	16	P	Bildschirm (GET, PUT)
FB	3000 - 0000	11	P	Feldbedeutung
FBS	3000 - 0000	11	P	Feldbedeutung für Feld mit Sperre
FE	3000 - 0400	00	P	Feldbedeutung
FBA	3000 - 1000	11	A	Feldbedeutung abdruckbar
FEA	3000 - 1400	00	A	Feldbedeutung erste abdruckbar
FAA	3000 - 2400	00	A	Feldbedeutung alle abdruckbar
HI	5800 - 0000	12	AP O	Hilfe
W	5800 - 0400	12	AP O	Weiter
I	0C20 - 0404	00	AP	Suchergebnis
S	0C20 - 0404	00	AP	Suchergebnis
F	0420 - 0018	00	AP	Fenster auf eine Spalte ignorieren
Z	0 20 - 1000	00	AP	ZPL
I	2020 - 0010	00	AP	Indirekt d.h. Verweisdateipointer
K	2020 - 0004	00	AP	Kette
*	2020 - 0008	00	AP	Null - Operation
SP	5020 - 1000	00	API	Sperre - rücksetzen
K ,K	2C04 - 0C00	00	A	Satz (Kapitel)
L ,K	1408 - 1800	00	AP	Satz (Kapitel)
KD	1408 - 1818	01	AP	Satz direkt
KS	1408 - 180C	00	API	Satz spezial (LEASY)
A	1408 - 1C04	06	AP	Abschnitt
F	1408 - 2008	01	AP	Feld
NI	08			Siehe: SUCHEN

Diverse Beschreibungen zu CIS

Kommando	Modul - UP#	KL	DIV.	Erläuterungen
OD	0C			Siehe: SUCHEN
OP	2008 - 0000	12	AP	Input
I	2008 - 0000	12	AP	Input
S	2008 - 0404	00	API	Spezial
U	2008 - 0004	12	AP	Update
PU, K	1400 - 0000	00	P	Satz (Kapitel)
A	1400 - 0404	00	P	Abschnitt
F	1400 - 0808	01	AP	Feld
KF	1400 - 2400	09	AP	Satz feldweise
AF	1400 - 2804	09	AP	Abschnitt feldweise
R	4000 - 0000	00	AP	Default Typ = Z
S	4000 - 0000	00	AP	Summarisch
T	4000 - 0000	00	AP	Tabellarisch = S
Z	4000 - 0000	00	AP	Zeilenweise
SE, S	041C - 0004	00	AP	Sequentiell
SP	501C - 0C00	00	AP	Sperre setzen
ES	501C - 1000	00	AP	Sperre rücksetzen
LJ	201C - 0008	00	API	LOGADR=J setzen
LN	201C - 000C	00	API	LOGADR=N setzen
G	201C - 0010	00	AP O	I/O Transformation TAB3 (X'400')
K	201C - 0014	00	AP O	I/O Transformation TAB2 (X'200')
TZ	201C - 0018	0A	AP O	Setzen Zwischentrenner
TE	201C - 001C	0A	AP O	Setzen Endentrenner
SI	0C00 - 0000	01	AP	EAM
E	0C00 - 0000	01	AP	EAM
I	0C00 - 0010	01	AP	VD - Pointer
V	0C00 - 0004	01	AP	VD
S	0C00 - 1800	12	AP O	Suchergebnis
SO	1000 - 0000	01	AP	Steigend
S	1000 - 0000	01	AP	Steigend
F	1000 - 0004	01	AP	Fallend
E	1000 - 0008	01	AP	Eindeutig
SP	5000 - 0000	00	AP	Sperren ZPL
Z	5000 - 0000	00	AP	Sperren ZPL
D	5000 - 1C00	00	AP	Sperren Datei
RA	5000 - 2000	00	API	Reservierung Anfang
RE	5000 - 2004	00	APIO	Reservierung Ende
SA	5000 - 2008	00	API	Secure Anfang
SE	5000 - 200C	00	APIO	Secure Ende
KD	5000 - 1400	01	AP	Sperre Satz direkt
SY	5000 - 2400	00	API	Synchronisieren (Test)
ST, SD	5000 - 1800	00	AP	Sätze / Datei
ST	5000 - 1804	00	AP	Sätze / TSN
SA	5000 - 1808	00	AP	Sätze / TSN + Datei
SG	5000 - 180C	00	AP	Sätze gesamt
DT	5000 - 1810	00	AP	Dateien / TSN
DG	5000 - 1814	00	AP	Dateien gesamt
DR	5000 - 1818	00	AP	Reservierte Dateien

Kommando	Modul - UP#	KL	DIV.	Erläuterungen
SU	0400 - 0000	01	AP	Suchen
L	0400 - 0000	01	AP	"LIMIT" mit Menge
S	0400 - 0001	01	AP	Mit Sperre
K	0400 - 0004	01	AP	Geklammert für sequentielles Folge - Oder
KS	0400 - 0005	01	AP	Geklammert für sequentielles Folge - Oder
A	0400 - 0008	01	AP	Abschnittsbezogen
AS	0400 - 0009	01	AP	Abschnittsbezogen mit Sperre
AV	0400 - 0010	01	AP	Feldinhalte (Aspektwerte) vergleichen
AVS	0400 - 0011	01	AP	Feldinhalte vergleichen mit Sperre
SY	2800 - 0000	00	AP O	SYS
TE	1808 - 1400	00	AP O	Kein Typ bei Text.
TR, A	4800 - 0000	00	AP O	Anfang
AL	4800 - 2000	00	AP O	Anfang - Lesen
AS	4800 - 3000	00	AP O	Anfang + Statusbehandlung
E	4800 - 0400	00	AP O	Ende
EB	4800 - 0800	00	AP O	Ende - bedingt
PE	4800 - 2404	12	AP O	Provisorisches Ende extern
R	4800 - 0C00	00	AP O	Rücksetzen
RB	4800 - 1000	00	AP O	Rücksetzen - bedingt
SA	4800 - 1404	12	AP O	Statusabfrage extern
SL	4800 - 1C04	12	AP O	Status löschen extern
UA	4800 - 1400	00	APIO	Statusabfrage
UC	4800 - 1800	00	APIO	Status CISKOR abfragen
UE	4800 - 2800	00	APIO	TR,E für UTM - Vorlauf
UL	4800 - 1C00	00	APIO	Stati löschen
UP	4800 - 2400	00	APIO	Provisorisches Ende (UTM)
UR	4800 - 2C00	00	APIO	TR,R für UTM - Vorlauf
WE	4800 - 2804	12	AP O	TR,E für externen Warmstart
WR	4800 - 2C04	12	AP O	TR,R für externen Warmstart
UN	04			Siehe: SUCHEN
V	7820 - 0800	01	AP	"Normales Join"
A	7820 - 0804	01	AP	Suchergebnis
F	7820 - 0808	01	AP	Forciert auch nicht vorhandene
AF	7820 - 080C	01	AP	Forciert auftragsbezogen
S	7820 - 0810	01	AP	Singulär MXN nicht ausmultipliziert
AS	7820 - 0814	01	AP	Suchergebnis nicht ausmultipliziert
FS	7820 - 0818	01	AP	Forciert nicht ausmultipliziert
AFS	7820 - 081C	01	AP	Auftragsbezogen/forciert/nicht ausmultipliziert
M	7820 - 0820	01	AP	Gemerkttes aus EAM verbinden
MF	7820 - 0828	01	AP	Forciert EAM Gemerktes
MS	7820 - 0830	01	AP	Gemerkttes nicht ausmultipliziert
MFS	7820 - 0838	01	AP	Gemerkttes/forciert/singulär
WA	4C00 - 0000	12	AP O	WARUM
WI	5C00 - 0000	12	AP O	WIEDERHOLE
WR, K	3C00 - 0000	02	AP	Satz (Kapitel)
KO	3C00 - 0008	02	AP	Satz mit Originalkey
S	3C00 - 0004	0A	AP	Selektiv
TS	3C00 - 0034	0A	AP	Selektiv mit TRANSFER - Vorspann

Diverse Beschreibungen zu CIS

Kommando	Modul - UP#	KL	DIV.	Erläuterungen
Z	1800 - 0808	09	A Z	Tabellarisch
Z	1800 - 0404	09	A Z	Zeilenweise
T	1800 - 0808	09	A Z	Tabellarisch
V	1800 - 0C0C	09	A Z	Verteilungstafel
VS	1800 - 0C00	09	A Z	Verteilungstafel mit Summen
G	1800 - 200C	09	A Z	Gruppensummen
GT	1800 - 2004	09	A Z	Gruppensummen tabellarisch
GS	1800 - 2000	09	A Z	Gruppensummen summarisch
A	1800 - 0000	00	A Z	Feldauskunft
B	1800 - 1010	00	A Z	Bildausgabe
N	1800 - 1820	00	A Z	Nächste Information
W	1800 - 1824	00	A I Z	Weiter
KX	1800 - 2404	00	A Z	Satz (Kapitel) hexadezimal
XT	1800 - 2408	09	A Z	Tabellarisch hexadezimal
TO	1800 - 2C08	09	A Z	Tabellarisch ohne Überschrift
TT	1800 - 2808	09	A Z	Tabellarisch ohne Überschrift
VK	3400 - 0004	01	A	???
M	2C00 - 8080	00	A	Maske
VB	3400 - 0008	01	A	VD - Balkendiagramm
E	0C00 - 1400	00	A O	ZPL - Keller
\$D	5400 - 0000	00	AP O	Diagnose

## 7.9 Liste der Module und Kommandos

Es folgt eine Liste der Module (nach Modulnummern sortiert) mit den behandelten Kommandos.

MODULNAM	Kommando	Modul - UP#	KL	DIV.
	BI	14		
	DR	04		
	NI	08		
	OD	0C		
	UN	04		
ISUCH	SU,AVS	0400 - 0011	01	AP
	AV	0400 - 0010	01	AP
	AS	0400 - 0009	01	AP
	SE,S	041C - 0004	00	AP
	I ,F	0420 - 0018	00	AP
	SU	0400 - 0000	01	AP
	L	0400 - 0000	01	AP
	A	0400 - 0008	01	AP
	KS	0400 - 0005	01	AP
	K	0400 - 0004	01	AP
	S	0400 - 0001	01	AP
IGET	G ,KPS	0800 - 0008	00	P
	KR	0800 - 0034	00	P
	N	0800 - 0014	00	P
	KNS	0800 - 001C	00	P
	KRS	0800 - 003C	00	P
	KP	0800 - 0000	00	P
	K	0800 - 0000	00	P
	KN	0800 - 0014	00	P
	A	0800 - 0400	06	P
	AS	0800 - 0408	06	P
	F	0800 - 0800	01	P
	FS	0800 - 0808	01	P
	NV	0800 - 0C14	00	P
	V	0800 - 0C0C	01	P
	ZP	0800 - 1000	10	P
	KDS	0800 - 1408	01	P
	KS	0800 - 1408	01	P
	KD	0800 - 1400	01	P
	KFRS	0800 - 183C	09	A
	KFR	0800 - 1834	09	A
	KFNS	0800 - 181C	09	A
	KFP	0800 - 1800	09	A
	KFN	0800 - 1814	09	A
	KFPS	0800 - 1808	09	A
	KI	0800 - 1C04	09	P
	KIS	0800 - 1C0C	09	P

Diverse Beschreibungen zu CIS

MODULNAM	Kommando	Modul - UP#	KL	DIV.
IZP	SI,E	0C00 - 0000	01	AP
		0C00 - 0000	01	AP
	V	0C00 - 0004	01	AP
	I	0C00 - 0010	01	AP
	I	0C20 - 0404	00	AP
	S	0C20 - 0404	00	AP
	AK	0C04 - 0400	01	AP
	Z	0C04 - 0400	01	AP
	I	0C04 - 0410	01	AP
	E	0C04 - 0400	01	AP
	F	0C08 - 0800	01	AP
	I	0C08 - 0810	01	AP
	E	0C08 - 0800	01	AP
	V	0C08 - 0804	01	AP
	I,Z	0C20 - 1000	00	AP
Z,E	0C00 - 1400	00	A O	
F,S	0C08 - 1804	00	AP O	
SI,S	0C00 - 1800	12	AP O	
ISORT	SO	1000 - 0000	01	AP
	E	1000 - 0008	01	AP
	F	1000 - 0004	01	AP
	S	1000 - 0000	01	AP
IUPD	PU,K	1400 - 0000	00	P
	A	1400 - 0404	00	P
	F	1400 - 0808	01	AP
	AE,KD	1404 - 0C10	00	P
	K	1404 - 0C00	00	P
	KS	1404 - 0C0C	00	PI
	A	1404 - 1004	16	P
	KF	1404 - 1408	09	AP
	F	1404 - 1408	01	AP
	L,K	1408 - 1800	00	AP
	KS	1408 - 180C	00	API
	KD	1408 - 1818	01	AP
	A	1408 - 1C04	06	AP
	F	1408 - 2008	01	AP
	PU,KF	1400 - 2400	09	AP
	AF	1400 - 2804	09	AP

MODULNAM	Kommando	Modul - UP#	KL	DIV.
IAUS	Z ,A	1800 - 0000	00	A Z
	Z	1800 - 0404	09	A Z
	T	1800 - 0808	09	A Z
		1800 - 0808	09	A Z
	VS	1800 - 0C00	09	A Z
	V	1800 - 0C0C	09	A Z
	B	1800 - 1010	00	A Z
	TE	1808 - 1400	00	AP O
	EN	1808 - 1404	00	AP
	Z ,N	1800 - 1820	00	A Z
	W	1800 - 1824	00	A I Z
	G	1800 - 200C	09	A Z
	GT	1800 - 2004	09	A Z
	GS	1800 - 2000	09	A Z
	XT	1800 - 2408	09	A Z
	KX	1800 - 2404	00	A Z
	TT	1800 - 2808	09	A Z
TO	1800 - 2C08	09	A Z	
IGETB	G ,B	1C00 - 0000	12	P
	P	1C00 - 0404	16	P
IVMZF	SE ,LN	201C - 000C	00	API
	G	201C - 0010	00	AP O
	K	201C - 0014	00	AP O
	TZ	201C - 0018	0A	AP O
	TE	201C - 001C	0A	AP O
	LJ	201C - 0008	00	API
	OP ,I	2008 - 0000	12	AP
	I ,I	2020 - 0010	00	AP
	K	2020 - 0004	00	AP
	OP ,U	2008 - 0004	12	AP
		2008 - 0000	12	AP
	I ,*	2020 - 0008	00	AP
	BL ,D	2004 - 0008	09	AP
	I	2004 - 000C	09	AP
	V	2004 - 0000	00	AP
	R	2004 - 0004	00	AP
C	2000 - 0000	12	AP	
A	2000 - 0004	00	AP O	
S	2000 - 0404	00	API	
OP ,S	2008 - 0404	00	API	
IVMLP	EX ,L	2404 - 0000	02	AP
ICMD	SY	2800 - 0000	00	AP O
EUPD	EI ,K	2C00 - 0000	00	A
	A	2C00 - 0404	00	A
	K ,K	2C04 - 0C00	00	A
	Z ,M	2C00 - 8080	00	A

Diverse Beschreibungen zu CIS

MODULNAM	Kommando	Modul - UP#	KL	DIV.
IGETF	G ,FB	3000 - 0000	11	P
	FBS	3000 - 0000	11	P
	FE	3000 - 0400	00	P
	FBA	3000 - 1000	11	A
	FEA	3000 - 1400	00	A
	FAA	3000 - 2400	00	A
IVSTAT	Z ,VK	3400 - 0004	01	A
	VB	3400 - 0008	01	A
IWRITE	WR ,S	3C00 - 0004	0A	AP
	TS	3C00 - 0034	0A	AP
	KO	3C00 - 0008	02	AP
	K	3C00 - 0000	02	AP
IRECH	R ,T	4000 - 0000	00	AP
		4000 - 0000	00	AP
	Z	4000 - 0000	00	AP
	S	4000 - 0000	00	AP
ICHKPT	TR ,A	4800 - 0000	00	AP O
	E	4800 - 0400	00	AP O
	EB	4800 - 0800	00	AP O
	R	4800 - 0C00	00	AP O
	RB	4800 - 1000	00	AP O
	UA	4800 - 1400	00	APIO
	SA	4800 - 1404	12	AP O
	UC	4800 - 1800	00	APIO
	UL	4800 - 1C00	00	APIO
	SL	4800 - 1C04	12	AP O
	AL	4800 - 2000	00	AP O
	PE	4800 - 2404	12	AP O
	UP	4800 - 2400	00	APIO
	UE	4800 - 2800	00	APIO
	WE	4800 - 2804	12	AP O
	UR	4800 - 2C00	00	APIO
WR	4800 - 2C04	12	AP O	
AS	4800 - 3000	00	AP O	
IWHY	WA	4C00 - 0000	12	AP O

MODULNAM	Kommando	Modul - UP#	KL	DIV.
ISPERR	SP	5000 - 0000	00	AP
	Z	5000 - 0000	00	AP
	SE,SP	501C - 0C00	00	AP
	ES	501C - 1000	00	AP
	I,SP	5020 - 1000	00	API
	SP,KD	5000 - 1400	01	AP
	ST,DT	5000 - 1810	00	AP
	DG	5000 - 1814	00	AP
	DR	5000 - 1818	00	AP
	SA	5000 - 1808	00	AP
	ST	5000 - 1804	00	AP
	SD	5000 - 1800	00	AP
	SG	5000 - 180C	00	AP
	SP,D	5000 - 1C00	00	AP
	SE	5000 - 200C	00	APIO
	RA	5000 - 2000	00	API
	RE	5000 - 2004	00	APIO
SA	5000 - 2008	00	API	
SY	5000 - 2400	00	API	
IDIAG	\$D	5400 - 0000	00	AP O
IHELP	HI	5800 - 0000	12	AP O
	W	5800 - 0400	12	AP O
IWIED	WI	5C00 - 0000	12	AP O
IADILOS	EX,A	7404 - 0000	02	AP
	TA	7404 - 0004	02	AP
IJOIN	AK,J	7804 - 0400	01	AP
	V,FS	7820 - 0818	01	AP
	AS	7820 - 0814	01	AP
	S	7820 - 0810	01	AP
	AF	7820 - 080C	01	AP
	F	7820 - 0808	01	AP
	A	7820 - 0804	01	AP
	M	7820 - 0820	01	AP
		7820 - 0800	01	AP
	AFS	7820 - 081C	01	AP
	MS	7820 - 0830	01	AP
	MF	7820 - 0828	01	AP
	MFS	7820 - 0838	01	AP



## 8 Diagnose - Routinen

### 8.1 Diagnose - Routinen in CISCON

#### 8.1.1 Protokollierung der IUTMDB - Schnittstelle

##### 8.1.1.1 Allgemeines

Es ist möglich die Informationen an der IUTMDB - Schnittstelle vor Verlassen von CISCON zu protokollieren. Somit enthalten die Bereiche die Werte nach der Bearbeitung durch CISCON und CISDBH.

Es wird auf SYSOUT protokolliert. Mit dem CMD - Makro wird das AID - Kommando %DISPLAY aufgerufen. Wird der Aufruf mit Fehler abgewiesen, wird das IDA - Kommando DISPLAY aufgerufen.

#### 8.1.2 Einschalten der Protokollierung

1. /SETSW ON=(14)

muß vor dem Starten gesetzt werden, da der Schalter nur einmal am Anfang abgefragt wird.

2. \$D CDIA J

kann jederzeit im laufenden Betrieb gegeben werden. Mit \$D CDIA N kann auch jederzeit wieder ausgeschaltet werden.

### 8.1.3 Protokollierung von CISCON

Bytes		Länge		Inhalt
dezimal	hex.	dez.	hex.	
0 - 0	0 - 0	1	1	OP - Code (von UTM gesetzt): 00 - Startparameter 04 - Connect to DBH 08 - Disconnect from DBH 0C - Asynchr. Disconnect 10 - User Call 14 - Finish DB Transaction 18 - Cancel DB Transaction 1C - Break DB Transaction 20 - Continue DB Transaction 24 - Status of DB Transactions 28 - Transaction in Peta
1 - 1	1 - 1	1	1	DBS (CIS interne Rückmeldung): 00 - Initialwert bei UTM - Call 01 - DBSCONDB - DBH connected 02 - DBSDISB - DBH disconnected 03 - DBSDWNDB - DBH down 04 - DBSCONTA - Transaction connected 05 - DBSDISTA - Transaction disconnected 06 - DBSRSTTA - Transaction reset 07 - DBSCLITA - Transaction close iss. 08 - DBSUPDTA - Transaction updated 09 - DBSTAUNK - Transaction unknown 0A - DBSTARST - Transaction reset 0B - DBSCONTR - Read - Transaction connected 0C - DBSPETA - Transaction in Peta 0D - DBSTAOP - Transaction open FF - DBSUCALL - User - Call - Initialwert
2 - 2	2 - 2	1	1	Secondary Opcode (1.Byte DBCONPAA) (Es können mehrere Bits gesetzt sein.) 01 - Wait for completion 02 - DC - system handles STXTS 04 - Check DB - status 08 - Delete DB - stati 10 - Coldstart & 1st task
3 - 3	3 - 3	1	1	Error Codes (2. Byte DBCONPAA) 00 - No errors 04 - DB - TA reset 08 - DB - TA chain reset 0C - DBMS down 10 - Administration error 14 - Try later 18 - Recoverable error 1C - Unrecoverable error 20 - User error

Bytes		Länge		Inhalt
dezimal	hex.	dez.	hex.	
4 - 11	4 - B	8	8	TSKM: 4 Bytes: Name des DC - Systems 1 Byte: Task - Typ: 00 - Batch 01 - Interaktiv 02 - TB 1 Byte: Status der DB - Verbindung: 80 - Connected 40 - Disconnected 20 - Abnormal disconnection 2 Bytes: Reserve
12 - 75	C - 4B	64	40	TAM: 30 Bytes: Daten für UTM 32 Bytes: Daten für CIS 2 Bytes: Reserve
12 - 12	C - C	1	1	Status der DB - transaction: (Es können auch mehrere Bits gesetzt sein.) 80 - DB - TA is open 40 - Close issued 20 - DB - TA closed 10 - DB - TA cancelled (Siehe Rollback - reason) 08 - Update - TA 04 - TA in Peta
13 - 13	D - D	1	1	UTM transaction control 80 - User close required 40 - Multi step TA allowed 20 - Keep task for multi step 10 - Announce task changes
14 - 14	E - E	1	1	Rollback reason 80 - User call 40 - DB administrator 20 - Deadlock 10 - Internal error 08 - Task deadlock 04 - Warmstart 02 - User error
16 - 40	10 - 28	24	18	UTM transaction id. 8 Bytes: Hostname 8 Bytes: Application name 8 Bytes: 1 Byte : Session counter 3 Bytes: Vorgang counter 2 Bytes: Transaction counter 2 Bytes: Vorgang number
16 - 23	10 - 17	8	8	Hostname. (Bei UTM - T: UTMtttt, mit tttt = TSN)
24 - 31	18 - 1F	8	8	Name der Anwendung (Siehe APPLNAME= im MAX - Parameter)
32 - 32	20 - 20	1	1	Session counter: Wird bei jedem Start einer UTM - session um 1 erhöht.
33 - 35	21 - 23	3	3	Vorgang counter Zähler für den Vorgang in einer session.

Bytes		Länge		Inhalt
dezimal	hex.	dez.	hex.	
36 - 37	24 - 25	2	2	Transaction counter Transaktionszähler innerhalb eines UTM - Vorgangs. Zu Beginn eines Vorgangs hat der Zähler den Wert 1.
38 - 39	26 - 27	2	2	Vorgang number Eindeutige Nummer in einer UTM - Anwendung. Jede Dialogstation bekommt bei der UTM - Generierung eine Vorgangsnummer zugeordnet, die sie so lange behält, bis eine neue Generierung der Anwendung vorgenommen wird. Zusätzlich bekommt jeder Task eine Vorgangsnummer mit der Generierung zugeordnet, die dann verwendet wird, wenn der Task ein Asynchronprogramm bearbeitet.
40 - 41	28 - 29	2	2	Reserve
42 - 49	2A - 31	8	8	Terminalkennung: (Siehe auch nächstes Feld) Aus KCLOGTER (LTERM) oder KCBENID (USER)
50 - 50	32 - 32	1	1	Kennung für vorhergehendes Feld: L - es enthält KCLOGTER B - es enthält KCBENID
51 - 51	33 - 33	1	1	Session counter (Wichtig für späteren Warmstart).
52 - 53	34 - 35	2	2	Reserve
54 - 57	36 - 39	4	4	TSN
58 - 63	3A - 3F	6	6	Aktuelles CIS - Paßwort (Spaces wenn kein Paßwort).
64 - 69	40 - 45	6	6	CIS - Paßwort beim letzten TR,A - Kommando (Spaces wenn kein Paßwort)
70 - 75	46 - 4B	6	6	Reserve
76 - 79	4C - 4F	4	4	Bei User Call: CIS - KL
80 - 107	50 - 63	28	1C	Bei User Call: CIS - K
76 - 107	4C - 63	32	20	Bei Startparameter: Parameter
108 - 111	64 - 67	4	4	Bei User Call: CIS - CM

### 8.1.4 UTM - DIAGAREA

In der DIAGAREA von UTM werden folgende Bytes belegt:

Bytes		Länge		Inhalt
dez.	hex.	dez.	hex.	
16 - 17	10 - 11	2	2	2 erste Bytes der CIS - Kommandos
18 - 18	12 - 12	1	1	Kommandoergänzung
19 - 19	13 - 13	1	1	DBS - Feld 01 - An DBH angeschlossen. 02 - Von DBH abgemeldet. 03 - DBH "down" 04 - Anfang Transaktion 05 - Ende Transaktion 06 - Transaktion zurückgestzt 07 - Ende für Transaktion 08 - Update - Transaktion 09 - Unbekannte Transaktion. 0A - Rückgestzte Transaktion. 0B - Anfang Transaktion nur zum Lesen. 0C - Transaktion in Peta. 0D - Transaktion noch offen.  FF - User Call
20 - 23	14 - 17	4	4	CIS - CM (Nur UTM - Vers. $\geq$ 3.0)
24 - 51	18 - 33	28	1C	CIS - Kommando (Nur UTM - Vers. $\geq$ 3.0)

## 8.2 Diagnose - Routinen in CISUTM

### 8.2.1 DIAGAREA

Die DIAGAREA ist ein Bereich in den CISUTM zyklisch Einträge schreibt. Den Bereich gibt es für jedes Modul CISUTM. Er enthält nur Daten aus diesem Task. Der Bereich hat einen Kopf von insgesamt 32 Bytes und 100 Einträge à 80 Bytes.

Anfang des Kopfes in CISUTM + X'4DEC'

#### 8.2.1.1 Kopf der DIAGAREA

Bytes		Länge		Inhalt
dez.	hex.	dez.	hex.	
0 - 15	0 - F	16	10	CISUTM ** DIAGAREA
16 - 19	10 - 13	4	4	Anfangsadresse
20 - 23	14 - 17	4	4	Endadresse + 1
24 - 27	18 - 1B	4	4	Adresse nächster freier Eintrag.
28 - 31	1C - 1F	4	4	Adresse aktueller Eintrag

#### 8.2.1.2 Eintrag in der DIAGAREA

Bytes		Länge		Inhalt
dez.	hex.	dez.	hex.	
0 - 7	0 - 7	8	8	Transaktionskennung
8 - 11	8 - B	4	4	Ursprungs - TSN
12 - 15	C - F	4	4	Bisherige CISUTM - TSN
16 - 16	10 - 10	1	1	Betriebsart
17 - 19	11 - 13	3	3	Reserve
20 - 23	14 - 17	4	4	Länge beim Laden der Daten (RLA).
24 - 27	18 - 1B	4	4	Länge beim Sichern der Daten (RSA).
28 - 59	1C - 3B	32	20	CIS - Kommando
60 - 63	3C - 3F	4	4	CIS - CM
64 - 71	40 - 47	8	8	Zeit Eingang CISUTM (STCK - Format)
72 - 79	48 - 4F	8	8	Zeit Ausgang CISUTM (STCK - Format)

### 8.2.1.3 Auswertung der DIAGAREA

Mit dem Programm CISDAMP kann eine mit DAMP (V2) erstellte Liste ausgewertet werden. Die Einträge der CISUTM - DIAGAREA werden abdruckbar aufbereitet. Das Programm kann auch eine Liste von mehreren Dumps auswerten. Anschließend kann die Liste sortiert werden (z.B. nach Zeit oder TID).

Beschreibung im Kopf des Sourceprogramms.

### 8.2.2 Fehlercodeabfrage in CISUTM

Im Modul CISUTM ist Platz für eine Tabelle von 32 Einträgen à 5 Bytes. Hier können die CIS - CMs "reingereppt" werden, bei denen CISUTM reagieren soll.

Feldname: TERR CISUTM + X'EEF1'

Jeder Eintrag hat folgendes Format:

4 Bytes: CM oder \* \* \* \* (= Alles)

1 Byte : Reaktionskennzeichen E: bei Fehler Dump diesen Task + weiter.  
D: bei Fehler Dump aller Tasks + weiter.  
T: wie D + Tasks beenden.

### 8.2.3 Protokollierung der CISUTM - Aufrufe

#### 8.2.3.1 Allgemeines

Es ist möglich die Aufrufe von CISUTM vor Verlassen des Moduls zu protokollieren. Somit enthalten die Bereiche die Werte nach der Bearbeitung durch CISUTM und CISI.

Es wird auf SYSOUT protokolliert. Mit dem CMD - Makro wird das IDA - Kommando DISPLAY aufgerufen. (Läuft noch nicht mit AID Kommando).

#### 8.2.4 Einschalten der Protokollierung

Die Protokollierung wird eingeschaltet indem der Schalter 14 vor dem Start des Programms gesetzt wird. Beim ersten Aufruf von CISUTM wird dieser Schalter geprüft, und dann nicht mehr.

Diese Logik kann ausgeschaltet werden, indem X'F0' an CISUTM + X'02AF' "gereppt" wird.

## 8.2.5 Protokollierung von CISUTM

### 8.2.5.1 Aufrufe

Jeder CISUTM - Aufruf wird vor Verlassen des Moduls protokolliert:

Bytes		Länge		Inhalt
dezimal	hex.	dez.	hex.	
0 - 7	0 - 7	8	8	Transaktionskennung
8 - 11	8 - B	4	4	Ursprungs TSN
12 - 15	C - F	4	4	Bisherige TSN
16 - 16	10 - 10	1	1	Betriebsart
17 - 19	11 - 13	3	3	Reserve
20 - 23	14 - 17	4	4	Länge beim Laden der Daten (RLA).
24 - 27	18 - 1B	4	4	Länge beim Sichern der Daten (RSA).
28 - 59	1C - 3B	32	20	CIS - Kommando
60 - 63	3C - 3F	4	4	CIS - CM
64 - 67	40 - 47	8	8	Zeit Eingang CISUTM (STCK - Format).
68 - 75	48 - 4F	8	8	Zeit Ausgang CISUTM (STCK - Format)

### 8.2.5.2 FCB - Infos

Bei jedem Schreiben und Lesen der FCB - Infos wird dieser Bereich protokolliert.

### 8.3 Diagnose - Routinen in CISUTMA

In CISUTMA gibt es die Möglichkeit im Line - mode eine zusätzliche Diagnosezeile angeben zu lassen. Dieser Modus wird eingeschaltet durch einen REP auf Adresse A. (Default - Wert ist C'N' = X'D5' =keine Diagnose - Zeile).

1. REP A,X'D1',OBJMOD=CISUTMA

Es wird folgende Zeile angefügt:

```
PEND=pp KCLOGTER=lllllllll KCBENID=bbbbbbbbb TIME=hhmmss
```

mit pp	PEND - Code
lllllllll	logischer Terminal - Name
bbbbbbbbb	Benutzer - Name
hhmmss	Zeit (Stunden, Minuten, Sekunden)

2. REP A,X'C3',OBJMOD=CISUTMA

Es wird folgende Zeile angefügt:

```
cccc vvv... tttt
```

mit cccc	CM
vvvv...	VM (Spaces bei CM = IMxx)
tttt	TSN

Diagnose - Routinen

#### **8.4 Diagnose - Routinen in CISDBH**

wird ergänzt.

#### **8.5 Diagnose - Routinen in CISKOOR**

wird ergänzt.

## 9 Diverse DSECTS

Es folgt eine Auflistung der wichtigsten DSECTS in CIS.

Liste der DSECTS:	POOLA	Pool - Anfang
	POOLS	Pool - Satz
	PKS	PK - Satz für CISDBH
	VMADD	VMADL plus VMEXT zentrale Adreßleiste
	KOPF	KOPF - Abschnitt des form. Auftrags
	ASPE	ASPE - Abschnitt des form. Auftrags
	ERGE	ERGE - Abschnitt des form. Auftrags
	ASPW	ASPW - Abschnitt des form. Auftrags
	KOMFLD	Kommunikations - Feld
	ZPLKOPF	ZPL - Kopf



## 10 Binderlisten